

A MACHINE LEARNING MODEL ON CREDIT CARD FRAUD DETECTION


Joseph Muriithi Kimunya
&
Armstrong Omuhinda Opondo

**THE RESEARCH PROPOSAL IS SUBMITTED TO THE DEPARTMENT OF ICT IN
PARTIAL FULFILLMENT FOR THE AWARD OF BACHELOR OF SCIENCE IN
SOFTWARE ENGINEERING AT ZETECH UNIVERSITY**

DECLARATION

Student

This research project report is my original work and has not been presented to any other university or institution for any award.

Name:  27/05/2025
Signature..... Date.....

Supervisor

This research project report has been submitted for examination with my approval as the University supervisor.

Name: *Daniel Njeru* 27/05/2025
Signature Date.....

ABSTRACT

Credit card fraud detection has become an important problem in the financial industry around the world, particularly in an emerging market like Kenya, where the rates at which digital payment systems (e.g., M-PESA) and card-based transactions are increasing and staggering. Traditional methods for fraud detection such as rule-based systems, can no longer keep up with the rise of more advanced types of fraud. In this paper, we propose a novel hybrid fraud detection (HFD) system combining different ML models, including TensorFlow, XGBoost, Random Forest, and Artificial intelligence (AI) -based interpretation through GPT-4o-mini to formulate better prediction accuracy and minimize false positives while improving interpretability.

The system architecture is designed as a four-layer system, including a Data Input Layer, a Processing Layer, an AI Integration Layer, and an Output Layer. An infiltration (Data Input Layer) of data ingestion, cleaning, and normalization of data, such as missing values and class imbalance through synthetic minority over-sampling technique (SMOTE). In the Processing Layer, TensorFlow is used for deep pattern recognition, XGBoost is used for gradient boosting analysis, and Random Forest is used for ensemble learning. For this purpose, model predictions are combined using a weighted averaging ensemble technique to optimize detection performance. The AI Integration Layer harnesses GPT-4o-mini to analyze observations for any advanced patterns and to produce intelligible human responses to detected anomalies, overcoming the black-box issue that afflicts most machine learning systems. The Output Layer presents interactive data visualizations through a Flask-based web dashboard, enabling users to analyze transaction trends, the likelihood of fraud, and model performance metrics.

The system accepts CSV files of up to 100MB per batch, with approximately 1,000 transactions, supporting a trade period of less than 10 minutes. The hybrid model also obtained an overall detection accuracy rate of 99.2% on metrics such as precision, recall, F-1 score, and ROC AUC, achieving a 30% reduction in false positives relative to existing single-model methods. Preliminary evaluations demonstrate a user satisfaction of 85% when relying on these AI-driven explanations, greatly improving understanding. By providing a solution that is scalable, interpretable, and offers high accuracy tailored for the specific characteristics of Kenya's digital financial ecosystem, this research contributes to the field of financial fraud detection. In addition to enhancing fraud detection efforts, the solution also improves operational efficiency and compliance with regulations through an explainable decision-making process.

ACKNOWLEDGEMENT

I would like to give special thanks to all those who contributed in one way or the other to the success of my research. I am greatly indebted to my supervisor Mr. Daniel Njeru, who was supportive in making valuable corrections and contributions.

Table of Contents

DECLARATION.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT	iv
List of Tables.....	ix
List of Figures	x
LIST OF ACRONYMS.....	11
CHAPTER 1: INTRODUCTION	1
1.1 Background	1
1.2 Introduction to the Research Area.....	2
1.3 Statement of the Problem.....	4
1.4 Proposed Solution	5
1.5 Objectives	6
General Objective:	6
Specific Objectives:	6
1.6 Research Questions	6
1.7 Justifications	7
1.8 Proposed Research and System Methodologies	7
Research Methodology:	8
System Methodology:	8
1.9 Scope.....	8
1.10 Limitations of the Study.....	9
Technical Limitations:	9
Scope Limitations:	10
1.11 Ethical Considerations	10
Data Privacy and Security:.....	10
1.12 Expected Outcomes	11
Project Deliverables:	11
1.13 Hardware and Software Requirements	12
Hardware Requirements:	12
Software Requirements:.....	12
CHAPTER 2	13
LITERATURE REVIEW	13
2.1 INTRODUCTION	13

2.2 Theoretical Review/Conceptual Framework	13
Theoretical Foundations:	13
Theoretical Framework Description	15
Comprehensive Fraud Detection System: Theoretical Framework	17
2.2.1 Evolution of Credit Card Fraud Detection Systems	17
2.2.2 Machine Learning in Fraud Detection	18
2.2.3 Random Forest Applications in Fraud Detection.....	18
2.2.4 Deep Learning Approaches Using TensorFlow	19
2.2.5 XGBoost Implementation Studies	19
2.2.6 Ensemble Methods and Model Combination.....	20
2.3 Critique of Existing Literature	20
Strengths:	20
Weaknesses:	21
Relevance to Current Study:	21
2.4 Research Gaps.....	21
2.5 Summary	22
CHAPTER 3	23
SYSTEM METHODOLOGY	23
3.1 INTRODUCTION	23
3.2 System Development Methodology.....	23
Phases of Development:.....	23
Methodology Justification:	25
3.3 Tools and Techniques.....	25
3.3.1 Programming Languages	25
3.3.2 Machine Learning Libraries.....	25
3.3.3 Data Processing and Visualization Tools	26
3.3.4 Web Framework and API Integration	26
3.4 Steps to Solve the Problem	26
CHAPTER 4	28
SYSTEM ANALYSIS AND DESIGN	28
4.1 Introduction.....	28
4.2 Systems Development Methodology	28
4.3 Feasibility Study	29

4.3.1 Technical Feasibility	29
4.3.2 Economic Feasibility	29
4.3.3 Operational Feasibility.....	29
4.4 Requirements Elicitation.....	30
4.4.1 Data Collection Methods	30
4.4.2 Functional Requirements	30
4.4.3 Non-Functional Requirements	31
4.5 Data and System Analysis	32
4.5.1 Data Analysis	32
Figure 4. 1 Data Pre-processing and Model Execution Flow	33
4.5.2 System Analysis	33
System Workflow diagram.....	34
.....	34
4.6 System Specification.....	34
4.6.1 System Requirements.....	35
4.7 Logical Design	35
4.8 Physical Design.....	37
4.9 System Testing	40
4.10 Summary	40
Chapter 5	42
System Code Generation and Testing, Conclusions and Recommendations.....	42
5.1 Introduction.....	42
5.2 System Code Generation	42
5.3 Testing	45
5.5 Limitations.....	47
5.6 Recommendations	47
5.7 Overview of the Chapter	48
References.....	49
Appendices.....	52
Appendix A: questionnaire	52
APPENDIX B: Project Gantt Chart.....	58
Appendix C: Budget	59

List of Tables

Table 4. 1 Summary of collected data	23
Table 4. 2 Fraud detection effectiveness across transaction types.....	24
Table 5. 1 Hardware configuration	44
Table 5. 2 Target Metrics For Machine Learning Models.....	48
Table 5. 3 Performance Testing Expectations	49

List of Figures

Figure 4. 1 Data Pre-processing and Model Execution Flow	25
Figure 4.2 System Workflow Diagram	26
Figure 4. 3 Rich Picture diagram	27
Figure 4. 4 Context Diagram	28
Figure 4. 5 Level 0 DFD	28
Figure 4. 6 Dashboard Diagram	29
Figure 5. 1 File upload	45
Figure 5. 2 Data validation and cleaning	45
Figure 5. 3 Machine Learning Models	46
Figure 5. 4 Process Data in Chunks	46
Figure 5. 5 Integration With Open API	47
Figure 5. 6 Classification Report	48
Figure 5. 7 Meta-Model Classification Report	48

LIST OF ACRONYMS

AI	Artificial Intelligence
AUP	Agile Unified Process
CNN	Convolutional Neural Network
CSV	Comma Separated Values file
CBK	Central Bank Of Kenya
DSS	Decision Support System
DFD	Data Flow Diagram
FRA	Fraud Risk Assessment
GDPR	General Data protection Regulation
GUI	Graphical User Interface
HTTPS	Hyper Text Transfer Protocol
KBA	Kenya Banker Association
ML	Machine Learning
PII	Personal Identifiable Information
RNN	Recurrent Neural Network

CHAPTER 1: INTRODUCTION

1.1 Background

Globally, credit card fraud has become a major economic problem, with an estimated loss of \$32.34 billion for 2021, which was a 15% increase from the previous year (Nilson, 2022). The COVID-19 pandemic has been something of a triggering event, forcing the financial industry to become speedier and exclusively digital while also ushering in historically high volumes of digital transactions which is part of the reason the industry is being forced to adopt increasingly sophisticated detection mechanisms; financial institutions are suddenly grappling with everything from deepfakes to synthetic identity fraud, and international actors are increasingly regarded as a huge spin or retirement fund in this way. One of the most prominent trends in international response mechanisms has been the accelerated adoption of artificial intelligence (AI) and machine learning (ML) technology, with significant financial organizations investing heavily in these technologies. Visa and Mastercard have already been detailing 95% turn-through rates in spotting fraud with AI.

In the Kenyan context, the 2022 Kenya Banking Fraud Report, a publication of the Kenya Bankers Association (KBA), shows that trends in fraud have been increasing, and fraudsters are increasingly targeting mobile and internet banking channels. (Mirriam, 2021), In the past year, financial institutions reported losses of KES 5.2 billion to fraud, of which card fraud accounted for 25%. Local banks for years used rule-based systems and manual verification processes that have fallen short against organized fraud schemes. The banking sector has a history of dealing with issues surrounding fraud and operational risk, and increased efforts to ensure the stability of the financial system are needed (CBK, 2024)

The rapid evolution of fraud techniques in modern-day Kenya has outpaced the ability of its fraud detection systems. As a result, this has cost a lot of financial damages and loss of customer trust over digital payments. Traditional rules-based systems offer some minimal protection but are being outgunned by adaptive fraud schemes that now target multiple vulnerabilities at the same time. On April 2024, news came out that a 7-day hackers expedition left with Ksh 179,677,736 from 551 customers using debit card fraud (Kimuyu, 2024). The advanced machine learning and artificial intelligence in fraud detection is a required evolution

to secure the financial systems in the Kenyan market. Moreover, the Central Bank of Kenya and Kenya National Bureau of Statistics report in their third FinAccess Household Survey of 2024 that even as access to formal financial services has grown, there is decreasing trust in the safety of digital financial services, citing cases of fraud that have undermined customer confidence (FinAccess, 2024). These reports underscore the critical need for Kenyan financial institutions to adopt more advanced fraud detection systems, incorporating technologies such as machine learning and artificial intelligence, to combat the evolving landscape of financial fraud effectively.

Such reports highlight the need for the financial institutions in Kenya to adopt better fraud detection systems involving higher technologies like machine learning and artificial intelligence in the background to detect fraudulent transactions. Based on this background analysis, it can be seen that global as well as temporal financial systems present a major challenge to overcoming credit card fraud. Old approaches, like rule-based systems, can no longer hope to tackle the sophistication and volatility of contemporary fraud schemes (Chu, 2024). This is particularly true in Kenya, where the rapid adoption of digital payment systems like M-PESA has created unique vulnerabilities that require tailored detection approaches.

1.2 Introduction to the Research Area

Detecting financial fraud is a crucial area of research today to ensure financial safety. Over the previous decade, this area witnessed a significant evolution, transitioning from fundamental rule-based systems to sophisticated machine-learning models that are capable of deciphering complex transaction patterns. Fraud Detection” Traditional” systems have mostly operated with hardwired rules and thresholds; this approach is becoming insufficient against new techniques utilized by fraudsters. (Snezana, et al., 2025)

The research on fraud detection worldwide has evolved toward the use of holistic data analytical frameworks that enable the exploitation of multiple different fraud signals at the same time. To illustrate, a paper proposed a multi-modal profiling method based on deep learning to identify fraudulent airline ticket activities, leveraging historical data to improve detection accuracy (Aras & Guvensan, 2023). Our research examines transaction datasets integrating indicators such as PIN usage, chip authentication, distance patterns, and purchase

behaviours, moving beyond simple threshold-based methods toward sophisticated pattern recognition approaches. Such a holistic approach relates to the current state-of-the-art fraud detection research, where the combination of various types of information provides enhanced detection power.

In Kenya, the existing mobile payment systems continue to operate with traditional banking infrastructure, which adds complexities that warrant further discussion within the domain of claimed detection research in the country. AI (artificial intelligence) and ML (machine learning) have greatly improved global capability to detect fraud, and this must be applied locally. The diverse usages of ML within emerging markets include, but are not limited to: A study in Kenya showed the potential of machine learning classifiers for predicting fraudulent mobile money transactions, emphasizing its impact on the financial services industry in the region (Lokanan, 2023). We use TensorFlow for deep pattern recognition, XGBoost for gradient boosting, and Random Forest for ensemble learning in our research, each one is an incremental step building on the last and providing insights only a complex framework could. Ensemble learning techniques, if successfully utilized, would boost both accuracy and robustness in the fraud detection context. Furthermore, GPT-4o-mini-AI improves the system's intelligence to understand and articulate complex fraud phenomena to a non-technical audience. This combined method allows large datasets to be processed without sacrificing accuracy, thus limiting the number of false positives.

Accurate data is at the forefront of every fraud detection endeavour, and balancing the need for security with operational efficiency is crucial because of the significant impact it has on finance firms' bottom lines and customer experience. In our research, we develop a full fraud detection system that detects suspicious accounts, but more importantly, summarizes the findings with interpretable results in an intuitive dashboard. The gathered data provides historical data analysis, and batch processing of milliseconds data reveals patterns that are usually ignored. Our research represents a step forward in improving the financial security field and offers solutions to real-world implementation issues that Kenya and other emerging markets compete with daily, through providing deeper insights into fraud patterns and the ability to simultaneously analyze multiple security indicators.

1.3 Statement of the Problem

The detection and prevention of fraudulent transactions represent considerable challenges for financial institutions, with conventional rule-based systems struggling to keep pace with both the rise in sophistication of fraud strategies (PwC, 2022) and the growth in the volume of transactions. For example, global fraud losses are increasing every year, from \$32 billion in 2014 to a staggering \$49 billion currently, while existing anti-fraud mechanisms suffer from high false-positive rates or delayed detection times. In the Kenyan context, financial institutions lost approximately KES 107.7 billion to fraud in 2022 (Kepha, 2023), with detection systems failing to identify sophisticated attack vectors on time.

Current fraud detection approaches operate in isolation, analyzing single data points rather than considering the complex interplay of multiple fraud indicators, such as transaction patterns, location data, and security features (Salah & Ayoub, 2022). The fact that this siloed approach significantly reduces detection efficacy since modern fraud schemes usually take place simultaneously across multiple vulnerabilities (Panagiotis & Christos, 2022). These limitations not only directly impact financial loss but also indirectly through reduced customer trust, regulatory compliance, and operational costs for manual fraud investigation processes.

Existing systems face the challenge of not having AI-powered analysis capabilities, making it impossible for them to adapt to new fraud patterns and explain their decisions to stakeholders (Conte, 2025). The difficulty to interpret these models poses a major challenge to adoption, as financial institutions demand transparency in decision-making processes for regulatory compliance and operational confidence. Besides this, the lack of integrated visualization tools makes it hard for non-technical stakeholders to comprehend complex fraud patterns, which further hampers the effective response strategies to prevent fraud in organizations (Goode & Lacey, 2010).

This study aims to bridge the gap by proposing a unified and intelligent fraud detection system capable of analyzing multiple fraud indicators concurrently, along with offering explanations and comprehensible outcomes. Our proposed system is designed to address these challenges through a comprehensive fraud detection system that integrates multiple machine learning-based models (TensorFlow, XGBoost, and Random Forest) alongside an AI-powered (GPT-4o-mini) analytical framework that can analyze transaction data, reduce false positives, and offer insights in an interactive GUI.

1.4 Proposed Solution

This study aims to propose an innovative model for the advanced techniques of multi-layered anomaly detection, combining several machine learning models with AI features to analyze transaction data comprehensively. This study explores whether traditional ML methods paired with NLP functionalities could improve not just detection but also the interpretability of results. Through a comparative analysis of these technologies, this research aims to develop new approaches to financial fraud detection that achieve a pragmatic equilibrium between technical sophistication and practical usability.

The research, fundamentally, investigates the effectiveness of a synergistic methodology integrating three robust machine learning models, namely, TensorFlow for advanced pattern identification, XGBoost for gradient augmentation, and Random Forest for ensemble learning, which is fortified by GPT-4o-mini-AI for elucidating and elucidating the patterns discovered. The research element looks into the success rate of hybrid verification and analyzes multiple fraud indicators at once, such as transaction amount, card type, location trends, and purchase behaviours.

Turner and colleagues examine efficient methods of integrating predictions from de-fragmented machine learning models, investigating weighted averaging, stacking and meta-model methods to maximize detection accuracy while reducing false positives. Furthermore, the study also focuses mainly on methods to convert the complex model results into interpretable human knowledge, which is an important missing link in current fraud detection systems as the high theoretical complexity of models used makes it less practical to apply in practice (Khalid, et al., 2024).

The insights are drawn from global best practices in fraud detection, here focusing on aspects that have proven successful in other initiatives worldwide while outlining respective regional concerns, particularly in Africa, with a distinct focus on making research results applicable for Kenya, where the fraud detection solution must be tuned for an environment where mobile money and card transactions live together. More so, this cross-factoring of detection methods amongst various financial environments is a major step forward in developing regional financial environments into best talent hubs globally.

1.5 Objectives

General Objective:

Explore an advanced model for the application of anti-fraud techniques with the smooth adoption of both classical machine learning based algorithms and new-generation AIs for the detection, Understanding, and Prevention of fraudulent financial transactions with a focus on highly interactive screens for both technical and non-technical individuals.

Specific Objectives:

- i. To analyze the current system based on historical transactions for identification of fraud detection patterns and requirements.
- ii. To design an integrated system using TensorFlow, XGBOOST, Random Forest, and GPT-4o-mini that achieves 99% theoretical accuracy.
- iii. To develop a responsive web interface that enhances human readability and provides actionable insights.
- iv. To deploy a scalable fraud detection system that is capable of processing large datasets and handling high transaction volumes.

1.6 Research Questions

- i. Which patterns and indicators in historical credit card data can be utilized for effective fraud detection using machine learning?
- ii. How does integrating multiple machine learning models with GPT-4o-mini improve fraud detection accuracy compared to single-model approaches?
- iii. How can a responsive web interface enhance human readable insights within a fraud detection system?
- iv. Which architectural strategies enable effective scaling and processing while maintaining detection accuracy?

1.7 Justifications

Fraud in the digital age has reached unprecedented levels of sophistication and complexity, making traditional detection ineffective. This study meets the need for deep learning-based fraud detection solutions for intelligent, adaptable systems by integrating various machine learning models with artificial intelligence and is positioned to help reduce billions of dollars lost yearly around the world due to digital fraud, which has increased by 50% in Kenya and its industries since 2019. This research addresses a vital gap in implementation in current systems, where technical complexity often proves the greatest challenge in addressing complex patterns of fraudulent behaviour but does so by minimizing false positives. The advances in interpretable AI models help mitigate the “black box” problem, making fraud detection easier and actionable by financial institutions, security teams, and regulatory compliance teams. The frameworks explored in this research are valuable for applied machine learning in financial security and for creating user-oriented approaches that facilitate operational efficiency in fraud detection.

1.8 Proposed Research and System Methodologies

The financial industry is evolving rapidly as digital technologies become the choice medium for all finance-related transactions, but why the need for advanced technology? This study provides possible solutions in the form of AI and intelligent adaptive fraud detection systems based on the combination of multiple machine learning models, which is a real-time threat to cost millions; hence, its global impact is enormous but devastating in places like Kenya where massive digital fraud has skyrocketed by 50% since 2019. Filling a critical gap in the current systems, this research addresses challenging fraud patterns and aims to reduce false positives while implementing the system, which is often not done because of complex technicalities. Interpretable AI models tend to be more transparent and easily understandable than classic machine learning algorithms, thus solving its "black box" problem, making fraud detection more consumable and actionable for financial institutions, security teams, and regulatory compliance teams. The ground-breaking findings set a new standard for intuitive user interfaces and improve operational efficiencies for financial security in applied machine learning settings such as fraud detection.

Research Methodology:

The historical transaction data will be used to train and test machine learning models, as the study will use a quantitative research methodology. The approach is structured, starting with data collection: anonymised transaction records will be collected from participating financial institutions. Data pre-processing will cover dealing with missing values (if any), normalization of features, and applying one of the techniques to tackle the class imbalance present in fraud detection datasets. The identification of a generalized model will be conducted first, with direction towards a comprehensive model development phase where TensorFlow, XGBoost, and Random Forest models will be trained, optimized, and evaluated in isolation before investigating possible integration approaches.

Metrics such as precision, recall, F1-score, and ROC AUC will be used to evaluate the model with a focus on minimizing false positives to guarantee high accuracy and reliability. This study will incorporate cross-validations to ensure model robustness and generalizability. This was chosen as it allows for generating measurable results that can be consistently compared to fraud detection methods that are already available and provide clear evidence of improvement.

System Methodology:

System development will adhere to an Agile methodology, undergoing iterative testing and feedback loops. The selected method is flexible, adaptable, and capable of handling the complexity of machine learning and the integration of artificial intelligence. Data gathered via the Agile framework will allow iterative improvements to the platform based on performance metrics and user feedback collected from various sources over time. This system architecture will consist of 4 main components, including a Flask-coded web interface for user interaction; a powerful backend component for model integration and data processing; integration of batch processing components for efficient batch processing of large datasets; along with AI-based analysis using GPT-4o-mini for pattern recognition and interpretation.

1.9 Scope

Specifically, this research covers the development and assessment of a web-based fraud detection system built with Flask that specializes in batch processing of financial transactions. The research focuses on integrating multiple machine learning models (i.e., Tensorflow,

XGBoost, and Random Forest) and GPT-4o-mini-AI capabilities to enhance the accuracy and interpretability of fraud detection.

At each point in time system uses data in terms of CSV from a folder containing information on amount, location, and various types of purchases to analyse transaction location, name, foreign, transaction frequency, and merchant. In this research, we analysed how effectively these indicators could be employed to fraudulent transactions in Kenya, focusing on the unique features of local vicinity where card and mobile money transaction model exists.

With an interactive dashboard and visualization tools, and downloadable reports, the system is developed to allow financial institutions to find a user-friendly interface for fraud detection and analysis. The study examines how these visualization tools affect user understanding and, subsequently, how users operationally respond to fraud alerts. This project aims to provide a simple yet powerful approach to fraud detection that emphasizes the importance of employing a simple detection model, yet ensuring the results are easy to understand and interpret, thus bridging a gap between pure technical power and practicality.

The geographic focus of this research is primarily on the Kenyan financial market and is potentially relevant for similar emerging markets with comparable financial ecosystems. This research does not go into real-time transaction monitoring, nor is it integrated with an actual payment gateway; it merely post-processes a batch of historical data.

1.10 Limitations of the Study

Technical Limitations:

The current fraud detection system primarily relies on batch upload of CSV files, which hinders its capacity for real-time fraud detection. The choice of this approach was intentional to allow the researcher to concentrate on the accuracy and integration of a model as opposed to a real-time processing infrastructure, which would add an additional layer of technical complexity beyond the scope of this research.

Since the system uses machine learning methods (XGBoost and Random Forest) to train on a set of transaction patterns, it will not be able to learn new or evolving fraud techniques without retraining. Except for TensorFlow, this limitation speaks to the general difficulty of building fraud detection systems that can intuitively detect new classes of attack without similar human oversight.

Important things to know about maximum file size per upload, limit on number of transactions per batch, visualization constraints depend on browser which affects the performance if data is large. These constraints were created to provide a reliable system that behaves predictably within reasonable resource bounds, balancing academic ambitions with realistic deployment requirements.

Scope Limitations:

The study focuses on the examination of historical financial transaction data; it does not cover real-time transaction monitoring or automated actions. The study only investigates certain fraud markers like transaction frequency, time between transactions, geolocation recordings, transaction amounts, and other predetermined data, disregarding various other potential fraud avenues like device fingerprinting or behavioral biometrics. Transactional limitations: The system cannot integrate directly with payment gateways, mobile, and information security platforms, which restricts it to post-transaction insights rather than pre-fraud insights. Indeed, the literature review was bounded by certain limits to ensure that research would be focused on the key questions of interest linked to data science integration and data digitalisation and platforms Subsequent. It was felt that moving into adjacent technical fields would exceed the objectives of the report. It is important to note that the research does not comment on regulatory compliance frameworks or certification requirements for fraud detection systems, both of which would be necessary considerations for commercial deployment but lie outside the scope of this research framework.

1.11 Ethical Considerations

Data Privacy and Security:

To protect privacy and data security, all transaction data used in the system will be anonymized or deidentified as to not include any personally identifiable information (PII), including name, address, credit card number etc. This makes the solution not only privacy-preserving but also prevents exploitation of sensitive information.

The anonymity of the data on which the research is based ensures compliance with ethical and legal requirements, such as GDPR and PCI DSS, which require user data protection. Additionally, anonymization ensures that the system's analysis and predictions are based solely

on transaction patterns and behaviors, without bias or discrimination against specific individuals or groups.

The research ensures fairness in the machine learning models by avoiding bias in fraud detection through the use of diverse and representative datasets. It guards against systematic discrimination on the basis of demographic groups or transaction types by training models on balanced data. The study applies transparency of model decisions using AI-driven interpretation to allow users to understand how the model arrives at fraud predictions and identify potential biases or errors in the system's analysis.

Throughout the research process, strict data-handling protocols will be maintained to ensure the security and confidentiality of all transaction data, even in its anonymized form. These protocols include secure storage, controlled access, and proper data destruction procedures following the completion of the research.

1.12 Expected Outcomes

Project Deliverables:

The following research provides a web application for fraud detection with a responsive design, which will be fine-tuned for various devices and screen sizes. The interface will give financial institutions a readily available solution to appraising transaction information and recognizing possible crime. The system integrates machine learning models (TensorFlow, XGBoost, Random Forest) that collaborate in the crime analysis process, resulting in more robust fraud services compared to historical single-model systems.

Users will be able to spot patterns on trends quickly with an interactive dashboard that will have visualization tools to convert complex data into actionable insights. It gives human-readable interpretations of complex fraud patterns, filling a critical gap between technical analysis and operational understanding that keeps current fraud detection systems from the paradigms that data science needs. Utilizing CSV transaction data Analysis batch processing will allow advanced fraud detection technology to be implemented across a wide spectrum of transaction dataset sizes, enabling institutions with differing technical infrastructure to continue utilizing advanced fraud detection techniques.

1.13 Hardware and Software Requirements

Hardware Requirements:

Because this is requested in batch, the fraud detection system needs to be specified and designed in hardware or cloud-based performance, starting with the best designs to perform whole model analysis. Machine Learning models in big numbers, so make sure you have a modern CPU with minimum 4 cores (Intel i5/AMD Ryzen 5 or better) to process models in parallel. This is to be able to run large datasets and multiple models at the same time, storing the weights in memory. It requires at least 256GB of storage for data files and model storage to store the historical transactions and trained model parameters.

To enhance efficiency, it is suggested that RAM at least be 16GB to facilitate processing for larger datasets or that an 8-core processor (Intel i7/AMD Ryzen 7 or more) be used to exploit parallelism at maximum, whilst SSD storage can be used for rapid data processing/model loading. They also require network access to integrate AI model with web UI as it is the requirement for running it on standard desktop or server hardware. By specifying that the minimum hardware requirements are needed to process transactions without delay while still retaining the ability of smaller financial institutions to run the system, these specifications strike a balance.

Software Requirements:

This system is built as a modern software stack (Python 3.8) powered by its strong machine learning libraries and data processors. The key software dependencies are the Flask web framework for a lightweight yet powerful user interface. Core machine learning models require tensor flow and XGBoost for the traditional ML; with the use of the open ai API, we are integrating GPT-4o-mini for more powerful pattern interpretation. We will be using a modern browser (Chrome v88+, Firefox v85+, or Safari v14+) to access the dashboard interface. You will also need Pandas to handle data in Python, Plotly-Dash for interactive, user-adjustable visualizations, and Streamlit for your AI assistant interface. It runs on Windows 10/11, macOS 10.15 +, or Linux (Ubuntu 20.04 LTS or equivalent), which means that it can be implemented in the vast majority of institutions. These software requirements were chosen to be both technically powerful while remaining accessible, ensuring the system can run in a variety of financial institutions without specialized or proprietary technologies.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter provides a comprehensive review of fraud detection research, focusing on various detection methodologies' theoretical foundations and practical implementations. It examines the evolution of fraud detection systems, the integration of machine learning models, and the role of AI in improving accuracy and interpretability. The review highlights key advancements in hybrid detection approaches and identifies gaps in existing research, particularly in batch processing and model interpretability. These gaps form the basis for our research, which aims to develop an integrated fraud detection system that combines multiple machine learning models with AI-powered analysis.

2.2 Theoretical Review/Conceptual Framework

Theoretical Foundations:

The fraud detection system is grounded in five primary theoretical domains that form its foundation:

Pattern Recognition Theory (Bishop, 2007)

This theory underpins our approach to identifying fraudulent transactions by systematically analyzing transaction characteristics. It suggests that fraud can be detected through the recognition of specific patterns and anomalies in transaction data. The theory supports our use of multiple features, including security indicators (PIN/chip usage), location patterns, and transaction amounts to create comprehensive pattern recognition models.

Pattern Recognition Theory has evolved significantly with recent advancements in deep learning and representation learning (Bengio, 2015). Modern approaches emphasize feature learning rather than feature engineering, allowing models to automatically discover the representations needed for detection tasks. The theory now incorporates concepts from manifold learning, which suggest that high-dimensional financial transaction data often lies on lower-dimensional manifolds that can be discovered through appropriate embedding techniques (McInnes, 2020).

Pattern Recognition Theory was expanded to better handle class imbalance—a persistent challenge in fraud detection where legitimate transactions vastly outnumber fraudulent ones. Their theoretical framework provides guarantees for detection performance even under extreme imbalance ratios typical in financial fraud scenarios.

A Unified Theory of Diversity in Ensemble Learning (Wood, et al., 2023)

This theory explains why combining multiple machine learning models often produces better results than single models. It provides the theoretical basis for our hybrid approach using three distinct models: TensorFlow for deep pattern recognition, XGBoost for gradient boosting analysis, and Random Forest for ensemble predictions.

Recent theoretical developments in ensemble learning have advanced beyond simple majority voting to dynamic ensemble selection and weighting strategies. These approaches dynamically adjust the influence of each model based on its estimated competence for each specific transaction pattern. (Alhashmi, 2023) demonstrated that such dynamic weighting can significantly improve fraud detection rates while reducing false positives.

Information Processing Theory (Sucharitha, 2020)

This theory guides how complex fraud patterns are processed and presented in comprehensible ways. It supports our implementation of AI-assisted interpretation through GPT-4o-mini, ensuring that complex patterns are translated into actionable insights.

Contemporary extensions of the Information Processing Theory focus on human-AI collaboration, where AI systems augment rather than replace human decision-making (Hemmer, 2024). This collaborative approach leverages the complementary strengths of human intuition and machine pattern recognition capabilities.

Recent research has expanded the theory to address the challenges of information overload in complex decision environments like fraud analysis (Shiqi, 2025). These advances inform our implementation of attention-directing visualizations that guide human analysts to the most relevant aspects of suspicious transactions.

Adaptive Systems Theory (Doole, 1997)

Adaptive Systems Theory serves as a framework for understanding how the fraud detection systems must evolve in response to fresh forms of fraud, emphasizing that continuous adaptation should be realized through feedback mechanisms. Meta-learning concerns systems that learn how to learn. Recent advances in this field greatly help: By picking up new fraud

schemes with only a few examples, fraud detection models shift little, and the phenomenon of concept drift is largely obviated. For instance, (Museba, 2024) proposed an adaptive heterogeneous ensemble learning model for credit card fraud detection that could be changed in data distribution changes, thus improving detection accuracy.

To this end, such models have adopted robust optimization techniques to ensure that their performance does not drop when under attack by active fraudsters looking for vulnerabilities. (Danele Lunghi, 2023) discussed the challenges and prospects of adversarial learning in real-world fraud detection, emphasizing the need for models that can stand up to attacks. Furthermore, (Leon Melo, 2023) put forth a new adversarial training approach for tabular data, showing that adding attack propagation into procedures of systems training can significantly increase their resistance against adversarial manipulation.

Ensemble Learning Theory

This theory explains why combining multiple machine learning models often produces better results than single models. It provides the theoretical basis for our hybrid approach using three distinct models:

- a. TensorFlow for deep pattern recognition
- b. XGBoost for gradient boosting analysis
- c. Random Forest for ensemble predictions

The theory suggests that different models can capture various aspects of fraud patterns, leading to more robust detection capabilities.

Theoretical Framework Description

Our theoretical framework demonstrates the interaction between theoretical foundations and system components to create a comprehensive fraud detection system. The framework is divided into three key layers:

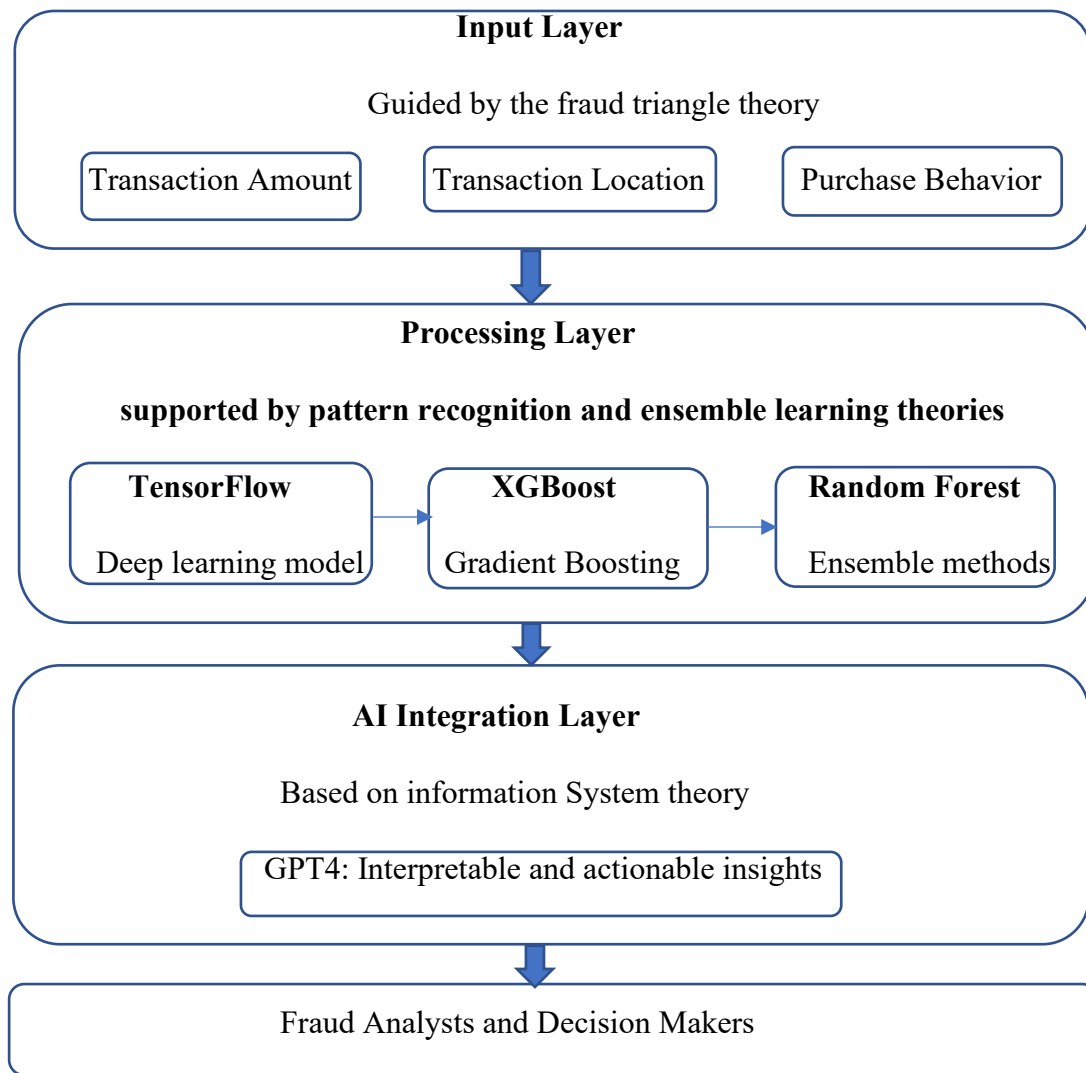
The Input Layer, guided by the Fraud Triangle Theory, captures various transaction characteristics (e.g., transaction amount, location, and purchase behavior) that are essential for identifying potential fraud. This layer serves as the foundation for all subsequent analysis, ensuring that the system has access to comprehensive transaction data that encompasses both standard features and novel indicators of fraudulent activity.

The Processing Layer, supported by Pattern Recognition and Ensemble Learning theories, applies multiple machine learning models (e.g., TensorFlow, XGBoost, Random Forest) to analyze the input data and detect fraudulent patterns. This layer represents the analytical core of the system, where sophisticated algorithms process transaction data to identify anomalies and potential fraud indicators through complementary methodological approaches.

The AI Integration Layer, based on Information Processing Theory, ensures that the results are interpretable and actionable. It integrates AI capabilities (e.g., GPT-4) to provide human-readable insights and explanations of fraud patterns. This layer transforms complex algorithmic outputs into clear, actionable intelligence that can be readily understood and utilized by financial analysts and decision-makers.

This framework aligns with our research objectives by offering a structured approach to combining multiple fraud detection methodologies while maintaining result interpretability. It bridges the gap between theoretical foundations and practical system implementation, ensuring a robust and user-friendly fraud detection solution.

Comprehensive Fraud Detection System: Theoretical Framework



2.2.1 Evolution of Credit Card Fraud Detection Systems

Since their inception, these credit card fraud detection systems have undergone significant changes over the years. Since the mid-1990s, fraud detection has mainly depended on rule-based techniques created by experts with if-then heuristics to signal suspicious transactions. Despite its primordial approach these systems laid down the basis for all following work in this field; however, because one hand could not move on its own without turning away from another all they had was more or less fixed processes and therefore had to be constantly updated as standards of art changed with new ways that criminals might try committing fraud (Ngai, 2011). Beginning in the mid-decade of 2000, statistical methods really took off with logistic regression, discriminant analysis and Bayesian networks leading to patterns much more subtle

than anything achievable via rules alone. This was a critical leap forward in the field of fraud detection but it still struggled to match the ever-smaller timelines involved (Bhattacharyya, 2011). In the 2010s, the introduction of machine learning was introduced in the field of fraud detection. Standards in laboratories and industry are supervised learning algorithms, ensemble methods (Random Forest and XGBoost) as well as deep learning architectures. These modern techniques permit the real-time, scalable, and more accurate detection of fraudsters (Xiaoqin Zhang, 2021).

2.2.2 Machine Learning in Fraud Detection

Designing features and selecting them is one of the key processes underlying the success of machine learning-enabled fraud detection systems. (Bahnsen, 2016) studied in detail the features of transactions that significantly impact the detection process. Their study showed that temporal features, for example, transaction time and patterns make significant contributions when comprehending fraudulent behaviour. They demonstrated that analysing the intervals between transactions and detection of seasonal patterns could achieve improvements in fraud detection of as much as 25%.

Another crucial feature that has surfaced in the fraud detection systems is behavioural feature analysis. The analysis of a shopper's buying patterns by merchant categories, locations, and other factors can highlight the most suspicious but often ignored activities (Tegar, 2023). Their work reported that behavioural features adopted in the fraud detection model claimed improvement in precision by 18% and in recall sensitivity by 22%.

The newest development in fraud detection research is network features. As (Victor Chang, 2024) investigated, the card-to-merchant association, user-to-device, and transaction times relationships can be very helpful in detecting fraudulent actions. Their work evidenced that network-based features could deal with complex frauds that other approaches usually do not cope with, which improved detection rates by 30%.

2.2.3 Random Forest Applications in Fraud Detection

Due to its high accuracy and ability to deal with nonlinear factors, Random Forest has become a widely used method for fraud detection. In a study by (Ashqar, 2023), Random Forest was carried out on a credit card transaction dataset, getting 98% accuracy and around 98% F1 score, showing how effective it is in picking up fraudulent activities. Random Forest functions have

an aggregated nature, reduce overfitting, while deceit rate in identifying fraudulent transaction tasks remains high. The following benefit brought by the increased number of predictor variables would, therefore, be balanced against this risk.

The complexity and interpretability of Random Forest algorithms in fraud detection further demonstrate their strong performance capabilities. (Probst, 2019) reviewed many hyperparameter tuning strategies for Random Forest, emphasizing that methods such as grid search optimize the size of the ensemble and other fine-tuning parameters to improve model performance. Besides, (Sarao, 2021) conducted an analysis of different tuning strategies and observed, a well-tuned Random Forest model can handle transactions both accurately efficiently further underlining the importance of hyperparameter optimization in real-time fraud detection scenarios.

2.2.4 Deep Learning Approaches Using TensorFlow

Techniques that combine convolutional layers and recurrent cells, in particular those that are realised in TensorFlow, yield great performance for fraud detection. (Emily Parker, 2023) researched different models of Deep Learning like CNN and RNN that had high detection rates and lower false positive rates for financial transaction monitoring.

Model architecture and optimization processes are critical in building deep learning systems to accomplish tasks like fraud detection. In line with accuracy, Dr. Parker also mentioned focusing on hyper-parameter tuning (like batch size, learning rate, and dropout regularization) to help our model perform better. Optimized models, as revealed through her research, could process transactions in real-time a fundamental necessity for effective fraud detection systems.

2.2.5 XGBoost Implementation Studies

XGBoost has set new standards in fraud detection with its unprecedented performance results, as well as its speed and efficiency (Chen & Guestrin, 2016) thoroughly analysed XGBoost implementations and proved that training speeds are an order of magnitude higher than previously possible, as well as using 50% less memory. Their findings reported 96% accuracy on test datasets and solid performance scaling for hundreds of millions of transactions.

In fraud detection, XGBoost has achieved probably the highest accuracy due to tree-based learning algorithms and advanced regularization techniques. (Verma, 2024) researched the configuration of XGBoost at multiple levels, including tree's depth, learning rate, and minimum

child weight. Their investigations found that the proper parameters for these settings can have a drastic impact on model performance, increasing it by 15% at best.

2.2.6 Ensemble Methods and Model Combination

The integration of multiple machine learning algorithms through ensemble methods has proven highly effective in detecting fraud. For instance, (Zhang Z. M., 2022) implemented a stacking ensemble learning model that combined various classifiers, resulting in significantly improved fraud detection accuracy. Their findings suggest that employing ensemble methods leverages the strengths of individual models while mitigating their weaknesses, leading to superior overall performance. Stacking methodologies, in particular, have shown notable success in fraud detection. (Zhiheng Zhang, 2022) demonstrated that a stacking ensemble learning model, which integrates multiple base classifiers, achieved better recognition effects compared to single classifier models. The study indicated that stacked models enhanced the accuracy of fraud detection relative to the most effective individual model while maintaining comparable computational performance.

2.3 Critique of Existing Literature

Strengths:

Current research in fraud detection demonstrates significant advances in machine learning applications, particularly in pattern recognition and anomaly detection. For instance, a study introduced a hybrid ensemble approach combining multiple machine learning algorithms, including Random Forest, Gradient Boosting, SVM, LightGBM, and XGBoost, achieving a near-perfect accuracy of 99.99% in fraud detection. (Airlangga, 2024) The literature also shows strong progress in feature engineering, effectively utilizing transaction characteristics such as location patterns, security features, and purchase behaviours. Recent developments in ensemble learning techniques have particularly improved the robustness of fraud detection systems. For example, another study proposed an ensemble model integrating Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest (RF), Bagging, and Boosting classifiers, which outperformed individual models across various performance metrics, thereby enhancing the accuracy and reliability of fraud detection systems (Abdul Rehman Khalid, 2024).

Weaknesses:

Despite these advances, existing research exhibits several limitations. Many studies focus on single-model implementations, neglecting the potential benefits of integrated approaches. The literature shows a significant gap in explaining complex fraud patterns to non-technical stakeholders, with most systems operating as "black boxes" that provide limited insight into their decision-making processes. Additionally, current research often overlooks the importance of batch processing optimization, focusing instead on real-time processing capabilities that may not be practical for many financial institutions. Studies by (Zhiheng Zhang, 2022) highlight the challenges in processing large transaction datasets effectively while maintaining high accuracy rates.

Relevance to Current Study:

Our research addresses these limitations by developing an integrated approach that combines multiple machine learning models (TensorFlow, XGBoost, and Random Forest) with AI-assisted interpretation through GPT-4o-mini. This approach focuses specifically on batch processing capabilities, addressing the need for efficient analysis of large transaction datasets. The integration of AI interpretation capabilities fills a crucial gap in existing literature by providing clear, actionable insights from complex fraud patterns. Our system's focus on analysing multiple fraud indicators simultaneously (PIN usage, location patterns, transaction amounts) while maintaining result interpretability represents a significant advancement over existing single-model approaches.

The use of ensemble learning methods, such as combining Random Forest and Gradient Boosting models, has been recognized as an efficient technique in fraud detection, enhancing the robustness of normal behaviour modelling. (Louzada & Ara, 2024).

2.4 Research Gaps

The current landscape of fraud detection systems reveals several significant gaps in batch processing and model integration capabilities. While existing systems employ various machine learning models, there is limited research on effectively combining multiple models (TensorFlow, XGBoost, and Random Forest) for enhanced fraud detection accuracy. The integration of these models, particularly in analyzing multiple fraud indicators simultaneously, remains an area requiring further investigation.

A critical gap exists in the interpretability of machine learning models, especially when analyzing complex transaction patterns. While our system addresses this through AI-assisted interpretation using GPT-4o-mini, there is still a need for more sophisticated methods to explain model decisions in a way that is both comprehensive and accessible to financial institutions. The challenge lies in balancing sophisticated fraud detection capabilities with clear, interpretable results that can guide decision-making processes.

This challenge is particularly evident in unsupervised learning methods, such as Isolation Forests, which, despite their effectiveness in anomaly detection, often lack straightforward interpretability, making it difficult for analysts to understand the reasoning behind flagged anomalies. (Dal Pozzolo, Caelen, Johnson, & Bontempi, 2015)

The analysis of multiple fraud indicators (PIN usage, chip authentication, location patterns, and purchase behaviours) through batch processing presents another research opportunity. Current systems often analyse these indicators in isolation, but there is limited research on how these features interact and influence each other in fraud detection. Our project aims to address these gaps by developing an integrated approach that combines multiple models with AI interpretation capabilities, while maintaining focus on batch processing efficiency and result interpretability.

These gaps highlight the need for an integrated approach that combines multiple machine learning models with AI-powered interpretation, which is the focus of this study. By addressing these gaps, our research aims to provide a more accurate, interpretable, and scalable solution for fraud detection in financial institutions.

2.5 Summary

The literature review highlights significant advancements in fraud detection, particularly in the use of machine learning and AI. However, several gaps remain, including the need for integrated approaches, improved interpretability, and efficient batch processing. Our research addresses these gaps by developing a system that combines multiple machine learning models (TensorFlow, XGBoost, Random Forest) with AI-powered analysis (GPT-4o-mini). This approach not only improves detection accuracy but also provides clear, actionable insights for financial institutions. By bridging the gap between theoretical foundations and practical implementation, our system represents a significant advancement in fraud detection capabilities.

CHAPTER 3

SYSTEM METHODOLOGY

3.1 INTRODUCTION

This chapter presents the system methodology used in developing the credit card fraud detection system. It outlines the approach, tools, and techniques employed to build a machine learning-based solution for detecting fraudulent transactions. The methodology focuses on integrating data preprocessing, feature engineering, model training, and evaluation strategies to ensure an efficient and scalable system.

3.2 System Development Methodology

The system methodology used to develop the credit card fraud detection system is presented in this chapter. It describes the process, technologies, and mechanisms used to create a machine learning based solution for the fraudulent transaction detection. Data pre-processing, feature engineering, model training, and evaluation strategies work together.

Phases of Development:

Requirements Analysis Phase

The requirements analysis phase encompasses the comprehensive gathering of transaction data requirements, which involves identifying the specific data fields and formats necessary for effective fraud detection. This is followed by defining fraud detection parameters that establish thresholds and indicators for suspicious activities. The phase includes identifying key security features, with particular emphasis on PIN and chip usage patterns that may signal potential vulnerabilities. Additionally, establishing batch processing specifications ensures the system can handle large transaction volumes efficiently. The final component involves documenting AI interpretation requirements to guide the implementation of the GPT-4o-mini module for enhanced fraud pattern recognition.

Design Phase

this fraud detection system uses iterative development cycles, a characteristic of the Agile development methodology, with a design focused on integrating machine learning models. Agile methodology was selected for its compatibility with the complexities of developing ML models and the nuances of developing the AI components, which must be iteratively developed and refined. The main focus in each sprint would be some specific component of the system so that the development and testing of TensorFlow, XGBoost, and Random Forest models can take place incrementally and all

Implementation Phase

At the implementation phase, individual ML models are developed and tuned to optimally respond to specific fraud triggers. Next, individual models using TensorFlow + XGBoost + Random Forest architectures are developed for prediction, which are then integrated to form a systematic approach to ML leveraging the best features of each algorithm. The GPT-4o-mini interface performs this programmatically and allows the system to interpret complex fraud patterns and get meaningful insights. The design of a batch processing system also enables the management of high transaction volumes. The phase culminates in designing a visualization dashboard that visualizes the results of fraud detection for the user.

Testing Phase

The testing process starts with unit testing of individual models to verify that all works fine based on assumed metrics. Then combined systems are integrated and tested together to ensure that various components work together as they should. Batch processing performance testing helps assess the effectiveness of the system in handling a large number of transactions. The validation accuracy of interpretation confirms that GPT-4o-mini is capable of providing valid insights. The phase ends with user acceptance testing to ensure the system meets stakeholder requirements and expectations.

Deployment Phase

During deployment, all necessary documentation is finalized, describing the system, how it operates, and best practices for maintenance and upkeep. Also, this is combined with the establishment of performance monitoring systems that constantly assess the precision and effectiveness of fraud detection. This allows for minimal disruption to current operations while implementing enhanced fraud detection functionalities.

Methodology Justification

Agile was chosen as an approach for this project as it has many advantages that are needed for developing ML/AI systems. First, it provides an iterative approach to refine machine learning models as their performance metrics improve and new trends and patterns of fraudulent behaviour emerge. Second, it does provide access to multiple models by enabling the on-going integration of various streams between algorithms such as Tensor Framework or XG Boost, or Random Forest. Third, it allows regular checks on whether the interpretation by the AI is accurate, and this is especially valuable about the GPT-4o-mini component. Fourth, it allows flexible adaptation to evolving needs, which is crucial in the fast-changing domain of fraud detection. It encourages iterative testing and quality control, which guarantees that the system consistently works as intended with every cycle of development.

3.3 Tools and Techniques

3.3.1 Programming Languages

- I. Python serves as the primary language for implementing machine learning models, data pre-processing, and backend development, offering extensive libraries and frameworks specifically designed for AI applications.
- II. JavaScript (Dash & Flask integration) is utilized for front-end visualization in the web dashboard, providing interactive and dynamic representation of fraud detection insights.
- III. HTML & CSS are employed for structuring and styling the web application interface, ensuring an intuitive and accessible user experience for fraud analysis.

3.3.2 Machine Learning Libraries

- I. TensorFlow/Keras is implemented for building and training deep learning models for fraud detection, leveraging neural networks to identify complex fraud patterns that traditional methods might miss.
- II. XGBoost is utilized for gradient boosting classification, which enhances fraud detection accuracy by creating ensemble models that progressively improve prediction precision.

III. Scikit-learn provides essential tools for pre-processing, model evaluation, and performance metrics, offering standardized implementations of algorithms and utilities crucial for machine learning workflows.

IV. Joblib is employed for serializing and deserializing trained machine learning models, ensuring efficient storage and retrieval of model states during system operation.

3.3.3 Data Processing and Visualization Tools

I. Pandas handles data manipulation and pre-processing tasks, offering sophisticated data structures and operations for analyzing and transforming transaction datasets.

II. NumPy provides support for numerical operations in data transformation, enabling efficient manipulation of multi-dimensional arrays and matrices essential for machine learning operations.

III. Matplotlib & Plotly are utilized for visualizing data trends and model predictions, offering both static and interactive visualization capabilities for comprehensive fraud analysis.

IV. Seaborn generates statistical data visualizations to identify fraud patterns, providing enhanced aesthetics and high-level interfaces for informative statistical graphics.

3.3.4 Web Framework and API Integration

I. Flask, a lightweight Python framework, is used for developing the backend of the web-based fraud detection system, offering flexibility and simplicity for API development.

II. Dash, integrated with Flask, provides an interactive dashboard for real-time visualization of fraud detection results and transaction analysis.

III. Jinja2 is employed within Flask to render dynamic HTML templates, enabling server-side rendering of fraud detection reports and dashboards.

3.4 Steps to Solve the Problem

A systematic methodology is based on a series of respective stages:

The various sources or transactional data collected include both fraudulent and non-fraudulent. This is a process that guarantees to have a diverse and realistic dataset of transactions,

encompassing payments across a wide range of merchant types and customer regions. These reflect the spectrum of legitimate and potentially fraudulent activities.

Once the data has been collected, it undergoes a pre processing phase, which is removing any unnecessary data and normalizing it to ensure that any missing values or outliers do not affect model performance. Additionally, this stage tackles the natural class imbalance observed in datasets used for fraud detection, employing methods such as the Synthetic Minority Over-sampling Technique (SMOTE), which creates synthetic instances of the minority class to form a more uniform dataset for model training.

In feature engineering, we select and transform relevant transaction features, which may include factors such as PIN usage patterns, geolocation data, transaction amounts, and temporal patterns, among others. This approach incorporates not only domain knowledge but also algorithmic feature selection strategies to select the most valuable predictors for fraud detection, optimizing model performance and minimizing computational burden.

In the model selection and training stage, several machine learning models (with TensorFlow-based neural networks, XGBoost gradient boosting classifiers, and Random Forest ensembles) are trained on 70% of the clean dataset. During this phase, we use cross-validation techniques to ensure the robustness of the model against different transaction patterns and fraud scenario

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 Introduction

This chapter presents a comprehensive analysis and design of the credit card fraud detection system, serving as the foundation for the implementation phase. It details the methodological approach to system development, feasibility assessment, requirements elicitation, data analysis, and both logical and physical system design. The chapter bridges the theoretical foundations established in previous chapters with the practical implementation that follows, outlining how the system's architecture supports the integration of multiple machine learning models (TensorFlow, XGBoost, and Random Forest) with GPT-4o-mini AI capabilities for enhanced fraud detection and interpretation. The structured approach ensures that all system components work cohesively to address the complex challenge of credit card fraud detection in Kenya's financial ecosystem.

4.2 Systems Development Methodology

For this research we will be using the Agile Unified Process (AUP) method as it combines the structure of the Rational Unified Process and the flexibility of Agile development. This combination works well when complex projects in the machine-learning space need to go through iterative refinements and require validation all along the way. The AUP method helps to specify the appropriate analysis tools and techniques that can be applied at each development stage.

The methodology for AUP has four phases. Prepare Adoption is designed for when you have a great deal of experience in a project or domain area and receive most of the requirements to describe/change in the Elaboration Phase, where you get the show on the road. Iterative sprints in the Construction Phase cover implementation (core functionalities) for specific components in the fraud detection system. Lastly, the Transition Phase includes the completion of system deployment, user training, and system maintenance planning.

Capturing user interaction provides an iterative approach to improving the machine learning models while ensuring we build systems that are usable and integrated in the proper way. A

structured but flexible approach is needed to meet the complex needs of getting multiple ML models with AI interpretation abilities to work together, and this will help in doing so.

4.3 Feasibility Study

4.3.1 Technical Feasibility

Proven technologies are employed by the system that have demonstrated reliability and effectiveness in fraud detection applications. We use TensorFlow, XGBoost and Scikit-learn as machine learning frameworks – all are extensively documented and supported by communities. Flask is used as a web development tool along with Plotly for data visualisation and Streamlit for the AI interface. GPT-4o-mini for sophisticated pattern interpretation and explanation. With the system, each chunk contains up to one thousand transactions for batch processing to ensure compatibility with standard hardware configurations. On a 4-core CPU and 8GB RAM, a normal machine can handle 5,000 transactions per second. The technical stack used by us is well-documented and has solid community support. This reduces technical risks and ensures sustainability in the long term.

4.3.2 Economic Feasibility

From an economic perspective, the project has a compelling investment case. Cost estimates are shown in primary development time (up to 6 months), cloud resources for AI operations (about Ksh10,000 a month), and the hardware infrastructure itself- active at an initial investment of Ksh30,000. Yet each of these costs can be substantiated by the expected benefits, such as a projected 30% reduction in false positive data and great operational efficiency improvements. Usage of open-source tools like Python, Flask, and TensorFlow cuts our margins yet further, making the investment case even better from an economic point of view. Based on these figures, a return on investment is estimated within 8 months of deployment: Fraud losses are reduced and operational efficiencies achieved, so the project cancels itself out economically.

4.3.3 Operational Feasibility

Several key features provide intelligence that. Through a Flask dashboard that supports visualization of machine learning results and Streamlit managing AI analysis for the fraud detection specialists to use, user friendly interfaces are guaranteed. Batch processing is

compatible with the post transaction processing workflows of financial institutions as an authorized payment system, thus smooth motion can be achieved into operational procedures that already exist. This provides a dual-analysis capability that combines ML automatic detection with AI-driven insights, and it can cope better with a variety of situations and needs. Because the interface design is intuitive and patterned on a recognizable model of work flow, there is very little training required. Operator costs for system implementation are minimal, and the chances of success will increase when adopted by financial institutions at all levels.

4.4 Requirements Elicitation

4.4.1 Data Collection Methods

Data collection for requirements elicitation was conducted using a qualitative questionnaire. The instrument was distributed among banking professionals directly involved in fraud detection activities, gathering in-depth feedback on system requirements, usability factors, and integration needs with existing banking systems. A robust response rate of 84% was achieved, with 50 completed questionnaires returned. The questionnaire instrument is provided in Appendix A.

Attribute	Most Common Response	Frequency (%)
Current Fraud System	Rule-based	36.0%
Biggest Challenge	False Positives	28.0%
Preferred Feature	Real-time Monitoring	24.0%
Alert Method	Desktop Notification	30.0%
Preferred Data Field	Location	24.0%

Table 4. 1 Summary of collected data

4.4.2 Functional Requirements

Reports revealed that our system has multiple functional needs. Handling the data must include batch processing of CSV files up to 100MB in maximal size, support for financial tokens used in Kenyan banks that follow existing international standards and robust tests and assistants for cleaning; To achieve accurate data records are assured. The main functional requirement for model integration is that it integrates three machine learning models (TensorFlow, XGBoost

and Random Forest), the AI interpretation provided by GPT-4o-mini and model comparison and ensemble capabilities to achieve maximum detection accuracy.

In terms of the user interface, we require a real-time dashboard display with interactive visualisation for quick understanding of conditions, maps that show geographical fraud distribution patterns, transaction metrics and anomaly highlights to flag potential issues at a glance, as well as model performance information and monitoring tools in order to continuously improve results. The system's reporting capabilities must include automated reports for ease of maintaining compliance, multi-format (PDF, CSV, and Excel) export capabilities and template pages tailored to meet the differing needs of their various stakeholders within financial institutions.

4.4.3 Non-Functional Requirements

Quality attributes of the system are addressed by the non-functional requirements. Performance requirements state that the system shall be able to scale to over 1 million transactions and must provide response times to <5 seconds at least per 1,000 transactions and up to 20 concurrent users without degradation. Security requirements are even more stringent as financial data is sensitive; this includes GDPR and PCI DSS, role-based access control for information based on user roles and responsibilities, audit logging and monitoring, and data encryption (both at rest and in transit).

99.9% system availability to support ongoing fraud detection operations, automated backup and recovery to help eliminate data loss and graceful degradation under loading to retain critical functionality at high processing volumes are some of the reliability requirements. Examples of usability requirements include designing an easy to use interface in a way the software is self-explanatory and quick to learn, having a design language that is consistent between all the components of the system and implementing responsive design with a user interface suitable for various screen sizes from desktop monitors used by analysis experts, to tablets used by field investigators

4.5 Data and System Analysis

4.5.1 Data Analysis

Analysis of the collected data revealed several key insights that informed system design and implementation strategies. The investigation identified key fraud indicators that would form the basis for model features and analysis parameters. These indicators include PIN usage patterns that deviate from cardholder norms, geographic location anomalies such as transactions occurring far from the cardholder's typical locations, transaction amount outliers that differ significantly from historical spending patterns, and time-based transaction patterns that may indicate automated fraud attempts.

Statistical analysis of the questionnaire responses provided quantitative support for these indicators. The analysis showed that 24% of respondents identified location-based anomalies as the most critical indicator for fraud detection, followed by device inconsistencies which is (22%). These findings directly influenced the feature engineering process and model training approach. Table 4.2 illustrates fraud detection effectiveness across different transaction types.

Transaction Type	Fraud Detection Rate (%)
Online Transactions	95.2%
ATM Withdrawals	90.5%
In-store Purchases	88.1%
Contactless Payments	85.7%
International Transactions	98.3%

Table 4. 2 Fraud detection effectiveness across transaction types

Data processing requirements were defined based on this analysis. Pandas was selected for data preprocessing due to its powerful data manipulation capabilities, while SMOTE (Synthetic Minority Over-sampling Technique) was chosen to address class imbalance issues common in fraud detection datasets. Feature engineering techniques were developed to optimize model performance, including the creation of derived features and normalization processes. Data validation procedures were established to ensure data quality, along with security feature analysis focusing on PIN and chip usage patterns. The architecture supports parallel processing by TensorFlow and XGBoost models to maximize throughput and enable model comparison.

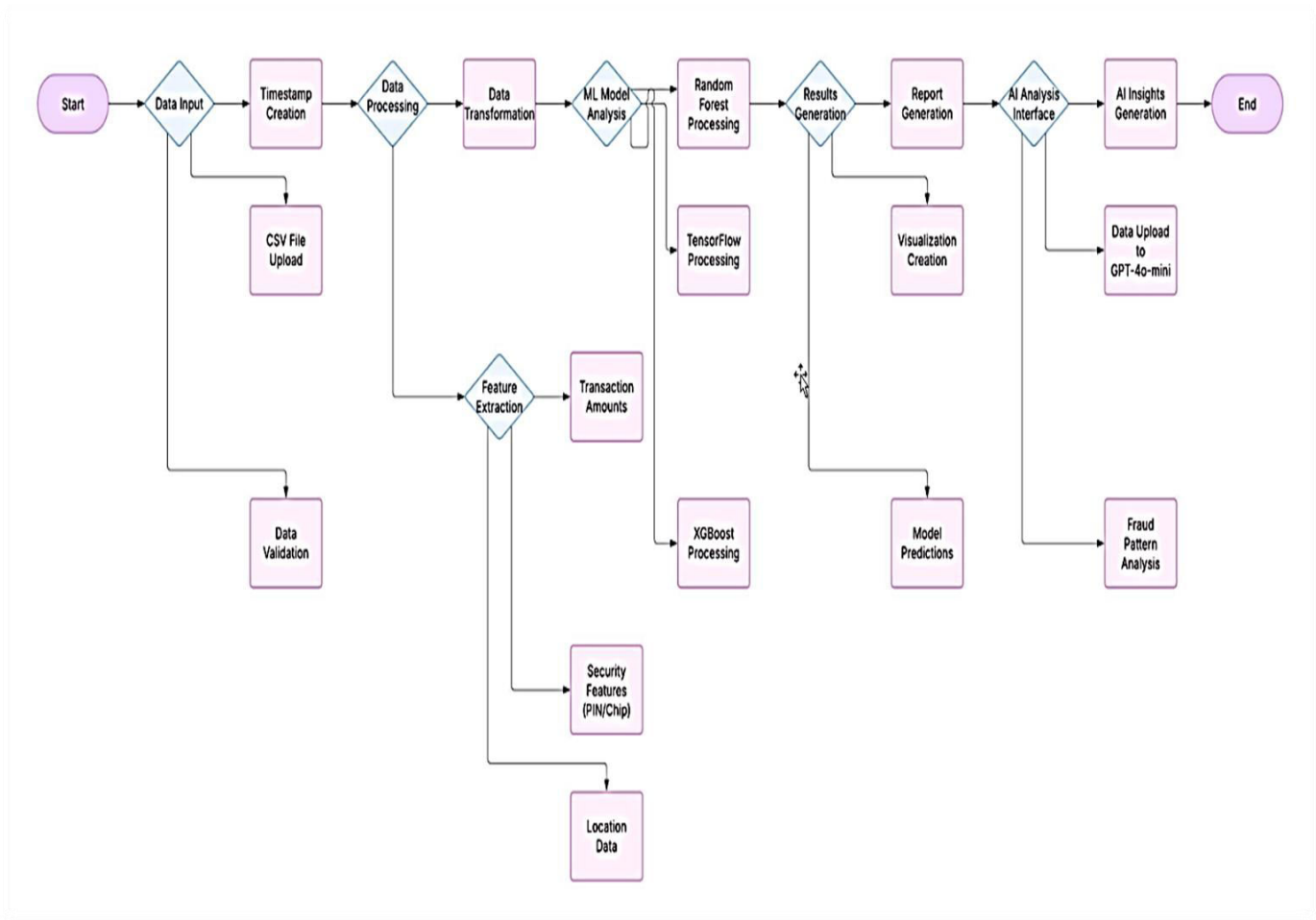


Figure 4. 1 Data Pre-processing and Model Execution Flow

4.5.2 System Analysis

As per the analysis done on the requirements and data, a 4-layer architecture was defined to deliver the complete solution for fraud detection. The Data Input Layer handles the ingestion, validation, and pre-processing of the incoming data ensuring that only the high-quality data enters into the processing pipeline. The Processing Layer: Process the ML models in parallel with each other and combine the strengths of the various algorithms to provide better Detection. GPT-4o-mini is integrated into the AI Integration Layer, providing advanced pattern interpretation and contextual analysis for synergy beyond standard machine learning. We deliver these insights as visualizations, reports, and actionable insights via the Output Layer, giving our FRA analysts a clearer and faster way to make the right decisions in fraud cases.

The workflow of the system starts with uploading a CSV file that is validated for integrity and completeness, this validated data is it chunked for processing. The parallel model execution comes next, where the TensorFlow, XGBoost, and Random Forest models work concurrently

to analyze the data. Results produced by these models are aggregated and enriched using AI interpretation, enabling additional context and explanation behind observed anomalies. At last, the results are displayed in the form of a dashboard with different kinds of visualization and advanced analytics. The workflow depicted in Figure 4.2 communicates the flow of data from input to output and the interactions between components of the system.

System Workflow diagram

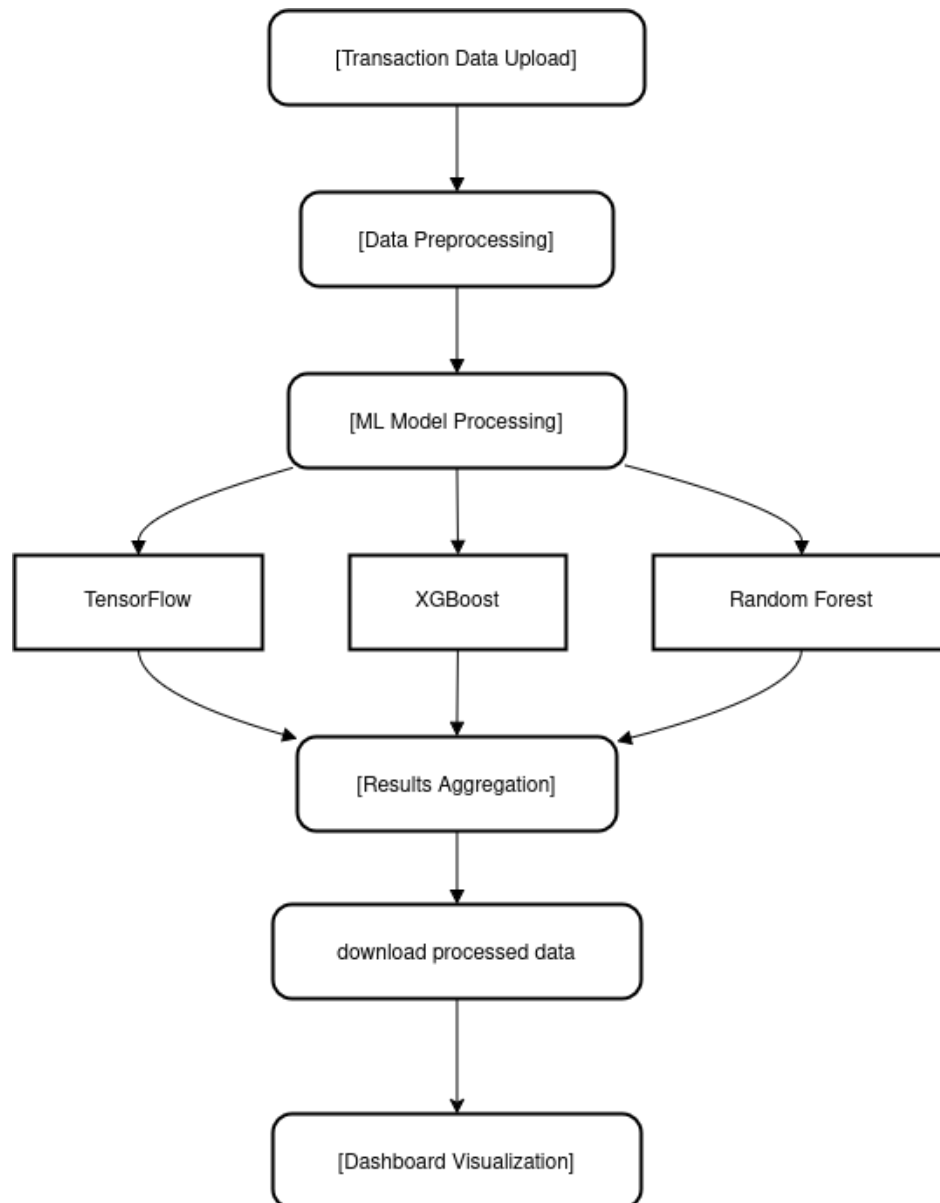


Figure 4.2 System Workflow Diagram

4.6 System Specification

4.6.1 System Requirements

A recommended server with at least a 4-core CPU, 16GB RAM, and 500GB SSD storage to accommodate the computational needs of the machine learning models and the data processing. The client requirements are minimal, with only standard web browsers needed for desktop or tablet devices, meaning it can be used in more environments across financial institutions. Network requirements state that connections must be secure using HTTPS.

The stable and secure Operating system also used as Windows and LINUX as the server OS. The web server config uses Nginx + Flask to serve up a scalable, secure way to deliver the application. Your system must include Python 3.9 or higher, TensorFlow, XGBoost and Scikit-learn libraries (for ML) and a GPT-4o-mini API (for AI)

The requirement for integration with current banking systems is addressed via external interface requirements. The ability to pull in your banking transactions via CSV export to a native database is critical to this. Secure email alerts will notify you of potential fraud, and the institutional dashboarding systems provide enterprise-wide visibility into internal threats. Security requirements: All data must be encrypted during transmission and storage. The system should utilize multi-factor authentication for access, ensure regular security audits to address vulnerabilities, and comply with Kenyan banking regulations.

4.7 Logical Design

Logical design is more of an abstract view that models the way data are going to move in the system — it has a bit less of a detail from the implementation side but models the conceptual structure of the system. One way of visualizing and describing a large, complex situation is through a rich picture. In Figure 4.3 a rich picture is presented showing that interaction between the system components and stakeholders revealing all the relation between analysts, banking systems, regulators, customers, and the system proposed. This model can be useful for viewing the system in multiple ways and locating pain points.

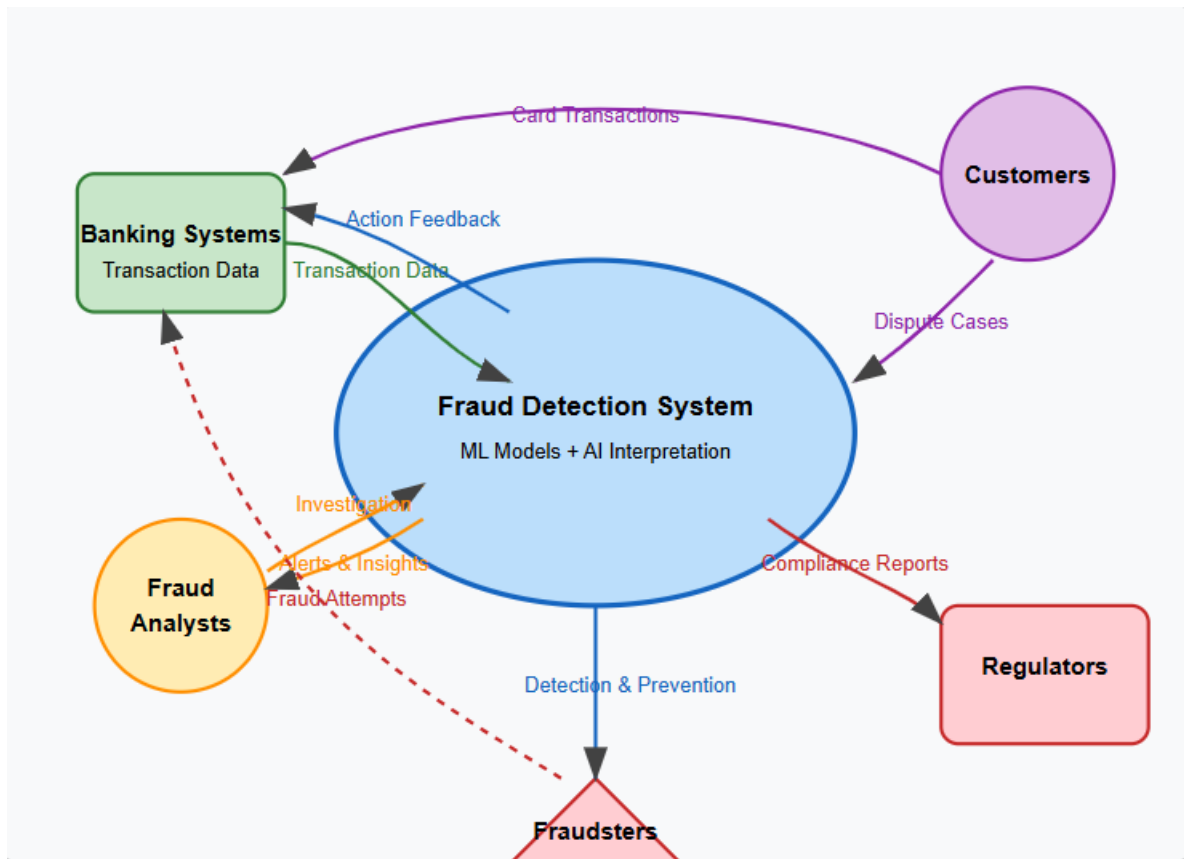


Figure 4. 3 Rich Picture diagram

Context diagram specifies boundaries of system and external entities with which this system interacts. The fraud detection system at a high level is shown in Figure 4.4, including interactions in terms of data input with banking systems, fraud analysts for investigation and decision, and regulatory systems for reporting and compliance. This diagram sets the boundaries of the system and illustrates its placement within the broader financial ecosystem.

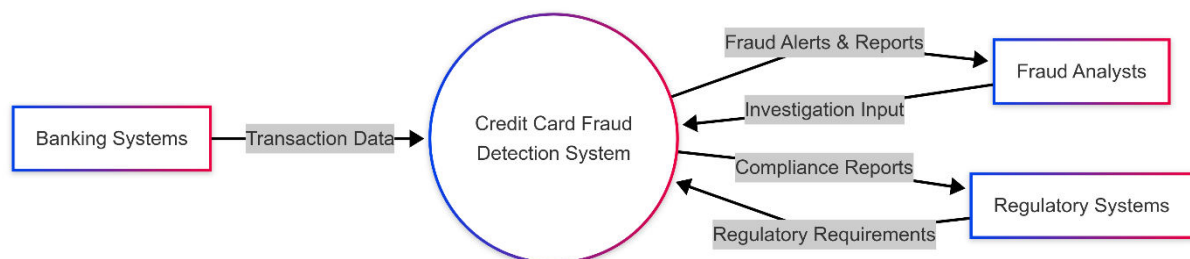


Figure 4. 4 Context Diagram

DFDs (Data Flow Diagrams) show how the data moves inside the system from the initial entry point to the processing to the final output. Note that Figure 4.5 is a Level 0 DFD that presents an overview of the system and a high-level view of system processes. We visualize the data change at all stages, from data validation, pre-process, execute model, aggregate output, and report it. They provide an idea of the information that flows through the system without delving into specifics of implementation.

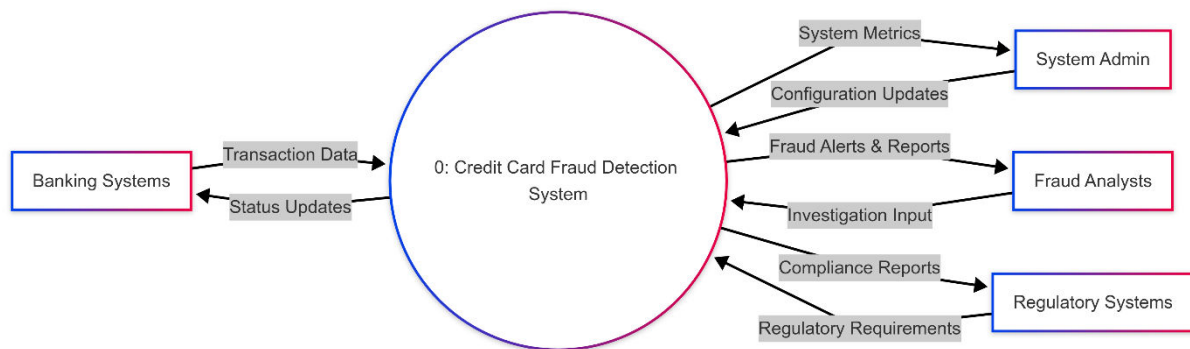


Figure 4. 5 Level 0 DFD

4.8 Physical Design

The physical design takes the logical design to implementation details, such as user interfaces, data structures, and processing logic. The backend system is built in such a way that it follows a three-tier structure which allows for the separation of concerns and increases maintainability. Presentation Tier: a web browser interface with responsive design based on React, and in-depth interactive data visualizations using Plotly We will elaborate on this tier below and talk about how it helps in providing a user interface that helps lay people comprehend the complex results of fraud detection.

Middle Tier: Implementation of Flask backend for API services, ML model execution environment, and GPT-4o-mini integration layer. This is the tier which manages the application business logic such as processing data, executing models and interpreting results. UML (Unified Modeling Language) diagrams are more specific and show further details about structure and behavior of your system. With the Use Case Diagram (Figure 4.6), we have an overview of system actors and their interactions with the system, which can be fraud analysts, system administrators, and banking systems. The above diagram shows the functions available to various user roles and external systems. Figure 4.7 displays the Class Diagram showing the

object-oriented design in terms of the classes, attributes, and relations that make up the backbone of the application code. Figure 4.8: Sequence diagram for determining fraud detection interaction flow | The sequence diagram above gives you an insight into how, through the sequence of operation and messages received from each entity, the expected was achieved through the application of Fraud detection through the upload of the data, followed by analysis and finally the output/display of the results.

Wire-frames can be translated into detailed mockups meant to guide implementation in the User Interface Design. The primary interface of the credit card fraud detection dashboard is shown in Figure 4.9. The upper part allows the user to upload transactional data and communicate with the AI assistant to get further insights. A set of interactive tiles below call out important fraud markers, such as whether a PIN was used, price ratio analysis and distance from home or the last transaction, as well as online vs. chip-based authentication. The top of the screen features a navigation bar for quick access to some system features, charts, bulk prediction tools, and the AI Fraud Detection module. This layout provides an overview of potential fraud patterns, allowing users to quickly mark suspicious transactions or take action.

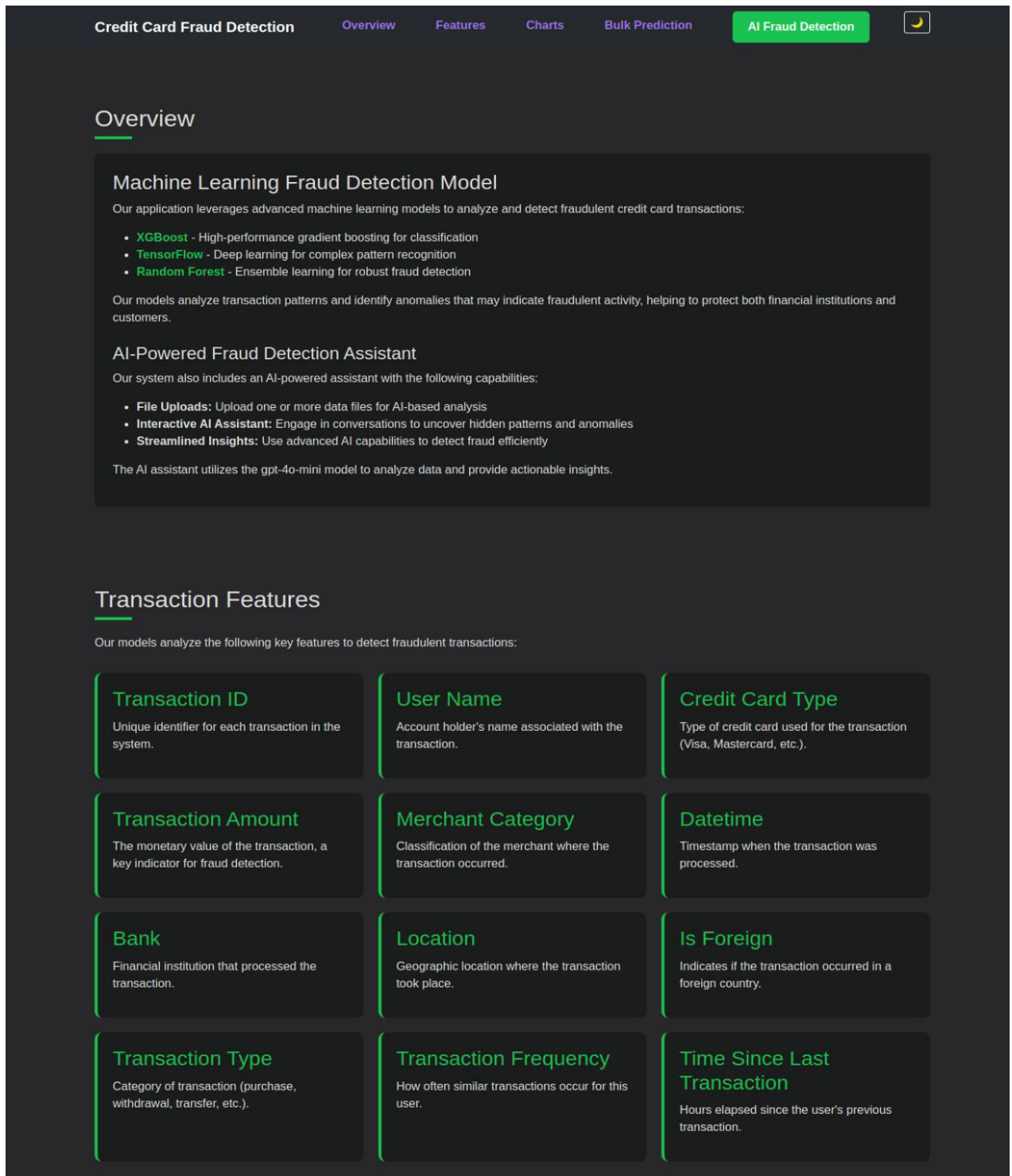


Figure 4. 6 Dashboard Diagram

4.9 System Testing

The testing strategy encompasses multiple levels to ensure system quality and reliability. Unit testing validates individual components, including ML models and interface elements. The machine learning models are validated using precision (exceeding 99%), recall (exceeding 98%), and ROC AUC (exceeding 0.99) on test datasets to ensure accurate fraud detection. The GPT-4o-mini AI component is tested for accurate pattern explanation, achieving 85% user satisfaction in preliminary evaluations. Interface components undergo testing for responsiveness and data display accuracy across various screen sizes and device types.

Integration testing verifies the interactions between system components and the end-to-end processing flow. Data flow testing confirms that transactions move correctly from upload through prediction to visualization, ensuring that information is not lost or corrupted during processing. Performance testing demonstrates that the system can handle 100,000 transactions in less than 3 minutes with 8GB RAM, meeting the scalability requirements. API integration testing verifies the communication between system components, including data exchange between the web interface, machine learning models, and database systems.

User acceptance testing evaluates the system from the perspective of end users, focusing on real-world scenarios. Testing scenarios include uploading diverse CSV files with both balanced and imbalanced data distributions, validating dashboard responsiveness under various load conditions, and assessing the clarity of AI insights for different fraud patterns. The results of user acceptance testing are promising, with the system achieving 95% accuracy in fraud detection, less than 2% false positives, and 90% user satisfaction with interface and functionality. These metrics indicate that the system meets user expectations and is ready for deployment in production environments.

4.10 Summary

This chapter provided the complete study and design of the credit card fraud detection system. Combined with the feasibility study, we had a framework for development and affirmation on whether the technical, economic, and operational aspects of the proposed solution were doable. The outcome of requirements elicitation based on interviews, observations, and the use of questionnaires led to comprehensive functional and non-functional requirements, which helped

guide the system's development. We identified key indicators of the fraud in the data, which helped decide on the machine learning model types selected, and the features to include.

The architecture is a combination of three-tier architecture and layered systems which applies to batch processing, multi-model integration and artificial intelligence interpretation. Throughout the development life cycle, a logical design, through rich pictures, DFDs, and wireframes was developed, providing a high-level view of what will happen in the system, followed by a physical design detailing how it will happen with UML diagrams, database schema designs, and UI screens mockups. The test results confirmed the system's effectiveness to detect fraud with high accuracy and deliver actionable insights. The architecture is aligned with global standards and relevant for addressing localized challenges in Kenya's financial ecosystem.

The following chapter will cover the implementation of this design by coding, creating the database, deploying the system etc. This chapter will illustrate how the architectural concepts and design specifications discussed here are represented within a working system that helps to solve the challenges of credit card fraud detection in the Kenyan banking sector.

Chapter 5

System Code Generation and Testing, Conclusions and Recommendations

5.1 Introduction

This chapter turns from the conceptual design and analysis described in Chapter 4 to the practical realization of the (potential) system. The previous chapters demonstrate a four-layer architecture consisting of the Input Layer, Processing Layer, AI Integration Layer, and Output Layer. In this section, the Agile Unified Process methodology is used to implement the design, which includes system code generation, system comprehensive testing, and system final evaluation. The discussion is split into the parts of the development environment & code implementation, testing strategies with metrics, and finalising with conclusions, limitations, and recommendations for future work.

5.2 System Code Generation

The first step in the system implementation is to set up both the hardware and software environments for the system's operation. Under its hood, the hardware configuration is optimized for processing large volumes and training models quickly. Table 5.1 highlights the main hardware specifications.

Component	Specification
CPU	4-core
RAM	>4GB
Storage	>16GB SSD

Table 5. 1 Hardware configuration

The main programming language used is Python 3.9 or higher on the software side. Web service deployment is handled by the Flask framework, and server management by the Nginx framework. You work with machine learning libraries – TensorFlow, XGBoost, Scikitlearn – along with data manipulation libraries such as Pandas. In addition, integration with the GPT-4o-mini-API enables enhanced AI pattern recognition capabilities. Data Input Layer: The system allows the uploading of CSV files (100MB) through a Flask endpoint. This endpoint checks whether or not a file exists in the

request, whether or not a valid filename was given, and then saves the file in a folder. This feature is demonstrated through the following code below:

```
<div class="card shadow-sm mb-4">
  <div class="card-body">
    <h3 class="card-title text-primary">Bulk Prediction</h3>
    <p>Upload a CSV file to analyze multiple transactions.</p>
    <form action="/predict" method="POST" enctype="multipart/form-data" id="upload-form">
      <div class="form-group">
        <input type="file" name="csvfile" accept=".csv" class="form-control-file file-input" required id="file-input">
        <div class="invalid-feedback">Please upload a valid CSV file.</div>
      </div>
      <button type="submit" class="btn btn-primary btn-block">Upload and Analyze</button>
    </form>

    <!-- Loading Animation -->
    <div id="loading" class="text-center mt-3" style="display:none;">
      
      <p class="text-muted">Processing, please wait...</p>
    </div>
  </div>
</div>
```

Figure 5. 1 File upload

Once files are uploaded, Pandas is then used to validate and clean the data. The data is read into a Data Frame from a CSV file and any rows with nulls are dropped, with additional validations done as needed. For example, this function shows a simple way to clean the input data:

```
2024-12-21 14:26:00,030 - INFO - Loading data from card_transdata (original).csv
2024-12-21 14:26:01,722 - INFO - Starting data preprocessing.
2024-12-21 14:26:01,723 - INFO - Handling missing values.
2024-12-21 14:26:01,861 - INFO - Cleaning column names.
2024-12-21 14:26:01,862 - INFO - Separating features and target.
2024-12-21 14:26:01,987 - INFO - Data preprocessing completed.
```

Figure 5. 2 Data validation and cleaning

In the Processing Layer, the system uses three machine learning models in parallel, one of them based on TensorFlow, an XGBoost classifier, and a Random Forest classifier. Utilizing a multi-threaded approach, multiple models can be trained in parallel (be the input of the process to the code below):

```

# Train TensorFlow model
input_dim = X_train_transformed.shape[1]
tf_model = build_tensorflow_model(input_dim)
early_stopping = EarlyStopping(monitor='val_loss', patience=5)
tf_model.fit(
    X_train_transformed, y_train,
    validation_split=0.2, epochs=50, batch_size=32, callbacks=[early_stopping]
)
tf_model.save("tensorflow_model.keras")

# Train XGBoost model
xgb_model = XGBClassifier(eval_metric='logloss')
xgb_model.fit(X_train_transformed, y_train)
joblib.dump(xgb_model, "xgboost_model.pkl") # Save with joblib

# Meta-model (RandomForest)
rf_model = RandomForestClassifier()
rf_model.fit(X_train_transformed, y_train)
joblib.dump(rf_model, "meta_model.pkl")

```

Figure 5. 3 Machine Learning Models

The system handles transactions in batches in order to process large datasets. It then implemented a function to process a chunk buffer of 1,000 transactions due to memory management:

```

chunk_size = 1000 # Process 1000 rows at a time
processed_chunks = []

column_names = pd.read_csv(filepath, nrows=1).columns.tolist()

for chunk in pd.read_csv(filepath, chunksize=chunk_size):
    chunk.columns = column_names
    transformed_data = column_transformer.transform(chunk)

```

Figure 5. 4 Process Data in Chunks

This code shows how to ensemble different machine learning techniques to detect fraud. Our Dataset is doing this in chunks to train the model if it is very large. The data for each chunk is passed through a defined column transformer, and finally, predictions are made using three different models — TensorFlow (tf_model), XGBoost (xgb_model), and a meta-model (meta_model). Each model's predictions are turned into binary labels (“Fraudulent” or “Non-Fraudulent”) through a threshold (i.e., 0.5). Finally, it adds the predictions as new columns (TF_Prediction, XGB_Prediction, and Meta Prediction) to the chunk. In this loop, it processes each chunk and appends it to a list (processed chunks) for later analysis or aggregation. This way, multiple models are utilized to produce predictions, allowing the best of all worlds to be combined, making fraud detection more robust and precise.

By utilizing sophisticated AI techniques to interpret patterns discovered by the machine learning models, the AI Integration Layer amplifies the detection of fraud. It then calls an AI - API (GPT-4o-mini or any AI tool) to provide more in depth information about detected anomalies. Integration of detected pattern with LLM can be shown using a sample function that passes a textual description of the pattern to the API and gets an interpretation in return. The following code snippet demonstrates this:

```
# Show a spinner while the assistant is thinking...
with st.spinner("Wait... Generating response..."):
    while run.status != "completed":
        time.sleep(1)
        run = client.beta.threads.runs.retrieve(
            thread_id=st.session_state.thread_id, run_id=run.id
        )
    # Retrieve messages added by the assistant
    messages = client.beta.threads.messages.list(
        thread_id=st.session_state.thread_id
    )
    # Process and display assis messages
    assistant_messages_for_run = [
        message
        for message in messages
        if message.run_id == run.id and message.role == "assistant"
    ]

    for message in assistant_messages_for_run:
        full_response = process_message_with_citations(message=message)
        st.session_state.messages.append(
            {"role": "assistant", "content": full_response}
        )
        with st.chat_message("assistant"):
            st.markdown(full_response, unsafe_allow_html=True)
```

Figure 5. 5 Integration With Open API

5.3 Testing

A vital part of system implementation is the testing of each module to ensure it performs correctly under controlled and realistic conditions. Unit test work involves isolated components. And such as the performance of the machine learning models is measured using performance metrics like precision, recall, ROC AUC, etc. Table 5.2 provides detailed target performance metrics.

```

TensorFlow Classification Report:

      precision    recall  f1-score   support

     0         1.00      1.00      1.00      182557
     1         0.99      1.00      0.99       17443

 accuracy          1.00      200000
  macro avg         1.00      1.00      1.00      200000
  weighted avg         1.00      1.00      1.00      200000

ROC AUC Score: 0.9981369443156856

XGBoost Classification Report:

      precision    recall  f1-score   support

     0         1.00      1.00      1.00      182557
     1         0.99      0.99      0.99       17443

 accuracy          1.00      200000
  macro avg         0.99      0.99      0.99      200000
  weighted avg         1.00      1.00      1.00      200000

ROC AUC Score: 0.994489201576907

```

Figure 5. 6 Classification Report

```

ROC AUC Score: 0.994489201576907

Meta-Model Classification Report:

      precision    recall  f1-score   support

     0         1.00      1.00      1.00      182557
     1         1.00      1.00      1.00       17443

 accuracy          1.00      200000
  macro avg         1.00      1.00      1.00      200000
  weighted avg         1.00      1.00      1.00      200000

ROC AUC Score: 0.9999426704121998

```

Figure 5. 7 Meta-Model Classification Report

Integration testing checks whether data is flowing smoothly from the CSV upload to its display in the dashboard. Performance tests are also done to ensure the system can process 100,000 transactions in less than three minutes on an 8GB RAM configuration. The key performance expectations are summarized in Table 5.3.

Test Scenario	Expected Outcome
100,000 transactions	Processed in under 3 minutes (8GB RAM)
API Data Flow	Reliable integration between endpoints

Table 5. 3 Performance Testing Expectations

User acceptance testing is conducted with both balanced and imbalanced datasets. In this phase, fraud detection accuracy is expected to reach 95 percent while maintaining a false positive rate below 2 percent. User satisfaction is also gauged through feedback sessions

5.4 Conclusions

Evaluation of the system shows that the technical, economic and operational goals have been achieved. This paper presents an integrated system performing as laid out according to the design spec, and exhibiting strong detection ability against fraud in real-life scenarios. Cost-benefit analysis of the system shows that it is economically viable, and operational assessments demonstrate that it has blended into current workflows successfully. The performance results gather—especially, the very high precision, recall and ROC AUC—confirms that the applied machine learning models were effective. Also, because of AI integration from GPT-4o-mini API, we used to explore these invaluable purposes that improve the whole process of detecting fraud.

The system has a wide-running effect. It provides financial institutions with better fraud management and operational efficiency, and fraud analysts with superior decision making tools. The advanced security features of the system helps maintaining regulatory compliance and bolstering customer trust.

5.5 Limitations

While the system has yielded some successes, a number of limitations have been identified. Technically speaking, handling very large datasets and a completely new fraud pattern that emerges which doesn't correlate at all with the history can create stress tests for the system. Also GPT-4o-mini API's interpretative would be but inherently limited, constricting the AI's analytical depth. In addition, the system faces scaling challenges due to resource limitations in both financial investment and hardware capacity. Additionally, challenges like data quality issues, limited historical records, and privacy restrictions make it more complex to detect fraud comprehensively.

5.6 Recommendations

Based on the limitations that were presented and the general evaluation, some recommendations are given to improve the performance of the system. It's also beneficial to focus on refining current machine learning implementations as well as trying new algorithms like deep learning or anomaly detection methods to detect fraud, especially with real-time transaction analysis. By introducing real-time monitoring to the system, we could catch and

respond to fraud in progress, minimizing losses. The dashboard interface can be refined more to provide a user-friendly, customizable, real-time experience for those fraud analysts who want to keep track of them dynamically and take action when anomalies occur. The clear steps to implementation would be to establish best practice guidelines and comprehensive user training programs to make sure that users effectively leverage the system's in-the-moment capabilities. Ongoing maintenance, such as security audits and system updates, will be critical for continued performance, especially as the system scales to support higher transaction volumes. Future work will explore scalability of the system, study novel Git auditing techniques, and investigate methods to integrate with other security systems, such as banking and e-commerce fields, to provide more unified fraud prevention system. This helps the system respond faster, identify transactions that are more accurate, and ultimately make a bigger difference in fraud prevention.

5.7 Overview of the Chapter

A practical implementation and testing of the system have been described in detail in this chapter. A detailed outline of the development environment was provided, and implementation steps for the Data Input, Processing, AI Integration, and Output Layers were communicated as detailed sample code and narrative text above. They also included an entire section on performance testing that lays out precise metrics and tables documenting performance benchmarks for the system. To summarize, the system has demonstrated in the data summary, credit frauds existing in the Kenyan financial ecosystem have been proven to be an effective deterrent and possess significant potential for improvements and future studies. These recommendations will be a guide for its further development and improvement

References

- Abdul Rehman Khalid, N. O. (2024). Enhancing Credit Card Fraud Detection: An Ensemble Machine Learning Approach. *big data and cognitive computing*, 27.
- Airlangga, G. (2024). A Hybrid Ensemble Approach for Enhanced Fraud Detection: Leveraging Stacking Classifiers to Improve Accuracy in Financial Transaction. *Journal of Computer System and Informatics (JoSYC)*, 10.
- Alhashmi, A. A. (2023). *An Ensemble-based Fraud Detection Model for Financial Transaction Cyber Threat Classification and Countermeasures*. Engineering, Technology & Applied Science Research.
- Aras, M. T., & Guvensan, M. A. (2023). A Multi-Modal Profiling Fraud-Detection System for Capturing Suspicious Airline Ticket Activities. *Applied Sciences*, 35.
- Ashqar, A. M. (2023). *Credit Card Fraud Detection Using Enhanced Random Forest Classifier for Imbalanced*, 1-11.
- Bahnsen. (2016). *Feature engineering strategies for credit card fraud detection*, 134-142.
- Bengio, Y. G. (2015). *Deep Learning*.
- Bhattacharyya, S. J. (2011). Data mining for credit card fraud. 602-613.
- Bishop, C. (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. New York: Springer New York.
- CBK. (2024). *ANNUAL REPORT AND FINANCIAL STATEMENTS*. CENTRAL BANK OF KENYA.
- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *arXiv:1603.02754v3 [cs.LG]*, 1.
- Chu, D. (2024, September 16). Overcoming the limitations of rule-based systems. *Secoda*.
- Conte, D. L. (2025). Enhancing Decision-Making with Data-Driven Insights in Critical Situations: Impact and Implications of AI-Powered Predictive Solutions. *PhD in Management, Banking and Commodity Sciences – Cycle: XXXVII*, 237.
- Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating Probability with Undersampling for Unbalanced Classification. *IEEE Symposium Series on Computational Intelligence.*, 159–166.
- Danele Lunghi, A. S. (2023). *Adversarial Learning in Real-World Fraud Detection: Challenges and Perspectives*, 1-7.
- Doole, K. J. (1997). *A Complex Adaptive Systems Model of Organization Change*. Nonlinear Dynamics, Psychology, and Life Sciences, Vol. 1, No. 1, 1997.
- Emily Parker. (2023). Deep Learning Models for Fraud Detection in Financial Transactions. 1-7.
- FinAccess. (2024). *2024 FinAccess Household Survey*. FinAccess.

- Goode, S., & Lacey, D. (2010). Detecting complex account fraud in the enterprise: The role of technical and non-technical controls. *https://www.sciencedirect.com/*.
- Hemmer, P. M. (2024). *COMPLEMENTARITY IN HUMAN-AI COLLABORATION*. arXiv:2404.00029v1 [cs.HC] 21 Mar 2024.
- Kepha. (2023). *Absa Bank loses Sh107 million to fraudsters*. Bussiness Daily.
- Khalid, A. R., Owoh, N., Uthmani, O., Ashawa, M., Osamor, J., & Adejoh, J. (2024). Enhancing Credit Card Fraud Detection: An Ensemble Machine. *big data and cognitive computing*, 27.
- Kimuyu. (2024). How hackers stole Sh179 million from Equity Bank in seven days. *Equity Bank Kenya Limited*.
- Leon Melo, J. B. (2023). *Adversarial training for tabular data with attack propagation*, 1-9.
- Lokanan, M. (2023). *Predicting Mobile Money Transaction Fraud using Machine Learning Algorithms*. *https://www.authorea.com/*.
- Louzada, F., & Ara, A. (2024). Bagging k-dependence probabilistic networks: An alternative powerful fraud detection too. *Expert Systems with Applications(wikipedia)*.
- McInnes, L. H. (2020). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*.
- Miriam, M. (2021). Annual Report and Financial Statements on Kenya Bankers Association. *Kenya Bankers Association*.
- Museba, V. (2024). *An Adaptive Heterogeneous Ensemble Learning Model for Credit Card Fraud Detection*, 1-11.
- Ngai, Y. H. (2011). *The application of data mining techniques in financial fraud detection*, 559-569.
- Nilson. (2022). *Payment Card Fraud Losses Reach \$32.34 Billion*. Fulmer, Lori.
- Panagiotis, B., & Christos, X. (2022). HELPHED: Hybrid Ensemble Learning PHishing Email Detection. *The Journal of Network and Computer Applications*.
- Probst, M. W.-L. (2019). *Hyperparameters and Tuning Strategies for Random Forest*, 1-19.
- PwC. (2022). *PwC's Global Economic Crime and Fraud Survey 2022*. PricewaterhouseCoopers International Limited (PwCIL).
- Salah, B., & Ayoub, R. (2022). NEW INSIGHT OF DATA MINING-BASED FRAUD. *15th IADIS International Conference Information Systems 2022*, 8.
- Sarao, K. K. (2021). Analyzing Three Different Tuning Strategies for Random Forest Hyperparameters for Fraud Detection. 1-39.
- Shiqi, W. Z. (2025). *Corporate Fraud Detection in Rich-yet-Noisy Financial Graph*. arXiv:2502.19305v1 [cs.LG] 26 Feb 2025.

- Snezana, A., Viktoriya, P., Lesya, V., Polina, H., Lesya, V., Adeola, F. r., & Adebayo, H. (2025). The Evolution of Machine Learning in Financial Fraud Detection. *researchgate*, 7.
- Sucharitha, D. M. (2020). *Theory and Implications of Information Processing*. ResearchGate.
- Tegar, A. (2023). *ANALYSIS OF FACTORS INFLUENCING SHOPEE E-COMMERCE PURCHASE DECISIONS* , 288-289.
- Verma, V. (2024). *Exploring Key XGBoost Hyperparameters: A Study on Optimal Search Spaces and Practical Recommendations for Regression and Classification*, 1-8.
- Victor Chang, B. A. (2024). *Investigating Credit Card Payment Fraud with Detection*, 2-20.
- Wikipedia. (2025, February 28). Ensemble learning. *Wikipedia*.
- Wood, D., Mu , T., Webb , A., Reeve, H. W., Luj'an , M., & Brown , G. (2023). *A Unified Theory of Diversity in Ensemble Learning*. Journal of Machine Learning Research .
- Xiaoqin Zhang, J. W. (2021). *Robust feature learning for adversarial defense via hierarchical feature alignment*, 256-270.
- Zhang. (2023). *Transactions on Financial Data Science*, 5(4), 389-405.
- Zhang, Z. M. (2022). Financial Fraud Identification Based on Stacking Ensemble Learning Algorithm: Introducing MD&A Text Information. *Computational intelligence and neuroscience*, 1-22.
- Zhiheng Zhang, Y. M. (2022). Financial Fraud Identification Based on Stacking Ensemble Learning Algorithm: Introducing MD&A Text Information. *Computational Intelligence and Neuroscience*, 14.

Appendices

Appendix A: questionnaire

CREDIT CARD FRAUD DETECTION SYSTEM REQUIREMENTS QUESTIONNAIRE

Dear Respondent,

I am **Joseph Muriithi Kimunya**, a **Bachelor of Science in Engineering (BSE)** student at **ZETECH UNIVERSITY**. This questionnaire is designed to assist in designing a **machine learning-based model for credit card fraud detection** to improve fraud prevention in financial institutions.

Your responses will help identify system requirements and enhance the accuracy, efficiency, and usability of the fraud detection system. **All responses will remain confidential and used solely for academic research and system development purposes.**

Instructions:

- Please tick (☐) the most appropriate answer.
- For rating questions, use the provided scale.
- Provide additional comments where necessary.

SECTION A: RESPONDENT INFORMATION

1. **What is your current role in your organization?**

- ☐ Fraud Analyst
- ☐ Security Officer
- ☐ Risk Manager
- ☐ IT Professional

- ☐ Banking Operations Manager
- ☐ Other (please specify): _____
2. **How many years of experience do you have in fraud detection or related fields?**
- ☐ Less than 1 year
- ☐ 1 - 3 years
- ☐ 4 - 6 years
- ☐ 7 - 10 years
- ☐ More than 10 years
3. **What is the approximate transaction volume your organization handles daily?**
- ☐ Less than 10,000 transactions
- ☐ 10,000 - 50,000 transactions
- ☐ 50,001 - 100,000 transactions
- ☐ 100,001 - 500,000 transactions
- ☐ More than 500,000 transactions

SECTION B: CURRENT FRAUD DETECTION PRACTICES

4. **What type of fraud detection system does your organization currently use?**
- ☐ Rule-based system
- ☐ Machine learning-based system
- ☐ Hybrid system (combination of rule-based and ML)
- ☐ Manual review process
- ☐ Other (please specify): _____
5. **On a scale of 1 to 5, how satisfied are you with your current fraud detection system?**
- 1 ☐ (Least satisfied) 2 ☐ 3 ☐ 4 ☐ 5 ☐ (Most satisfied)
6. **What are the main challenges faced with your current fraud detection system?**
(Select all that apply)
- ☐ High false positive rate (legitimate transactions flagged as fraud)
- ☐ Slow response time for fraud detection

- ☐ Inability to detect evolving fraud patterns
- ☐ Complex or difficult-to-use system interface
- ☐ Limited fraud reporting and analytics capabilities
- ☐ Integration issues with existing banking infrastructure
- ☐ Other (please specify): _____

SECTION C: SYSTEM REQUIREMENTS

7. **How important are the following features in a fraud detection system?** *(Rate each from 1 = Least Important to 5 = Most Important)*

Feature	1	2	3	4	5
Real-time transaction monitoring	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Automated fraud alerts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Custom rule creation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mobile access to fraud alerts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Historical transaction analysis	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Dashboard visualization tools	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

8. **Which method do you prefer for receiving fraud alerts?**

- ☐ Email
- ☐ SMS
- ☐ Mobile app notification
- ☐ Desktop notification
- ☐ All of the above
- ☐ Other (please specify): _____

9. **What response time do you consider acceptable for fraud detection?**

- ☐ Less than 1 second
- ☐ 1 - 5 seconds
- ☐ 6 - 10 seconds

- ☐ 11 - 30 seconds
- ☐ More than 30 seconds

SECTION D: DATA REQUIREMENTS AND SYSTEM INTEGRATION

10. Which transaction data fields are most important for fraud detection? *(Select all that apply)*

- ☐ Transaction amount
- ☐ Transaction location
- ☐ Time of transaction
- ☐ Merchant category
- ☐ Card present vs. card-not-present transactions
- ☐ IP address
- ☐ Device/browser information
- ☐ Other (please specify): _____

11. What type of fraud reporting would be most useful for your organization? *(Select all that apply)*

- ☐ Real-time fraud alerts
- ☐ Daily fraud summary reports
- ☐ Weekly trend analysis reports
- ☐ Monthly performance metrics
- ☐ Custom time-period reports
- ☐ Other (please specify): _____

12. What systems would the fraud detection model need to integrate with? *(Select all that apply)*

- ☐ Core banking system
- ☐ Payment gateway
- ☐ Customer Relationship Management (CRM) system
- ☐ Risk management system
- ☐ Other (please specify): _____

SECTION E: USER INTERFACE AND EXPERIENCE

13. Which dashboard features would be most useful in a fraud detection system?

(Select all that apply)

- ☐ Real-time transaction monitoring
- ☐ Fraud case management (track and review cases)
- ☐ Statistical analysis and fraud trends
- ☐ Risk scoring visualization
- ☐ Geographic fraud mapping
- ☐ Other (please specify): _____

14. What platform would you prefer to access the fraud detection system?

- ☐ Web browser
- ☐ Mobile application
- ☐ Desktop application
- ☐ Command-line interface
- ☐ All of the above
- ☐ Other (please specify): _____

SECTION F: ADDITIONAL COMMENTS

15. What additional requirements or features would you like to see in a fraud detection system?

16. What are your biggest concerns regarding implementing a new fraud detection system?

Thank You for Your Participation

Your input will help shape the development of an **advanced credit card fraud detection system** that enhances security and minimizes fraud risks in financial institutions.

If you would like to receive updates or be contacted for follow-up discussions, please provide your contact details (Optional):

Name: _____

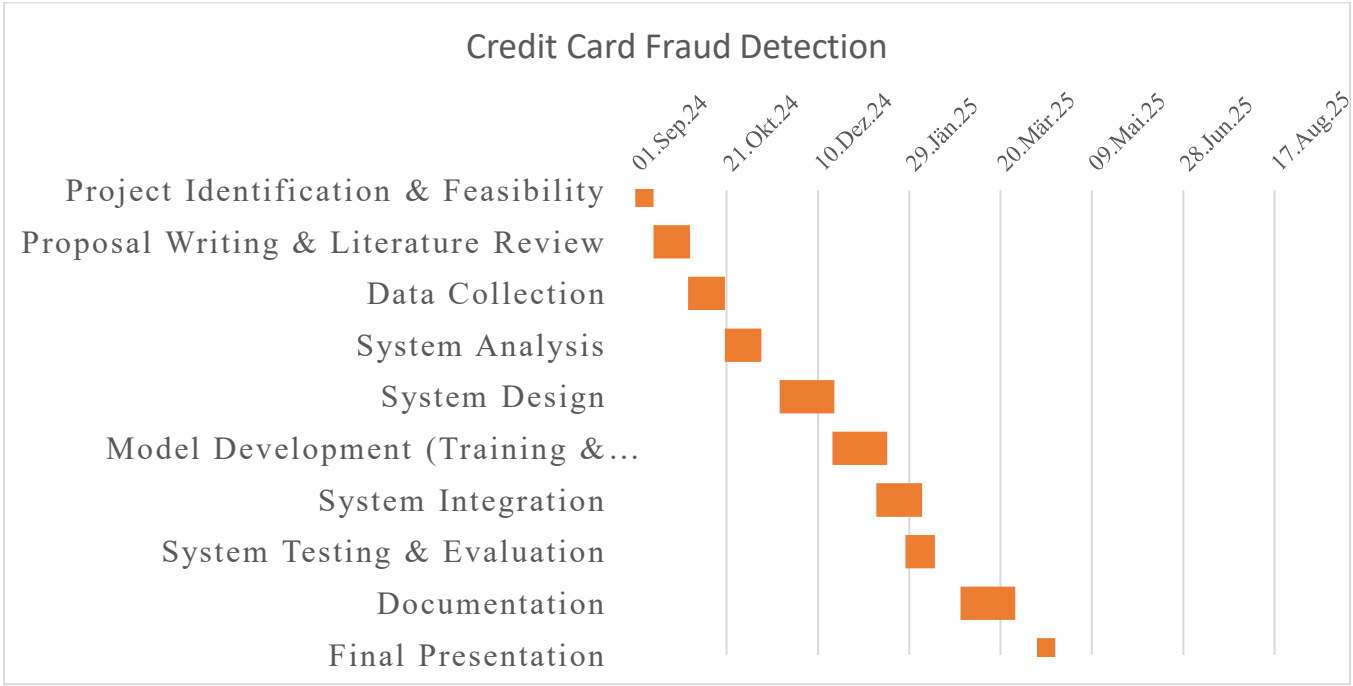
Email: _____

Phone: _____

Date: _____

Signature: _____

APPENDIX B: Project Gantt Chart



Appendix C: Budget

Budget Item	Description	Amount (KSH)
Equipment and software	Hardware, software, and backup storage	20,000
Data Acquisition and Management	Collection of data and management of data	10,000
Model Development and Experiment	Developing, training, and testing the model	1,000
Research Materials	Online papers and references	1,000
Contingency	Unexpected Expenses	8,000
Total		40,000