Machine Learning Model for Real-Time Digital Banking Fraud Detection

By

Allan Kiplagat Kemboi

169595

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology at Strathmore University

School of Computing & Engineering Sciences

Strathmore University

Nairobi, Kenya

June, 2025

This thesis is available for Library use on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Declaration and Approval

Declaration

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

© No part of this thesis may be reproduced without the permission of the author and Strathmore University

Student's Name: Allan Kiplagat Kemboi

Sign _____ Date 2025-05-31

Approval

The thesis of Kemboi, Allan Kiplagat was reviewed and approved for examination by the following:

Dr. Nelson Ochieng,

School of Computing and Engineering Sciences,

Strathmore University

Dr. Julius Butime,

Dean, School of Computing & Engineering Sciences,

Strathmore University

Prof. Bernard Shibwabo,

Director of Graduate Studies,

Strathmore University

Abstract

Fraudsters use financial fraud to deceive for the purpose of financial gains. It has now become an international threat for companies and organizations. In Kenya, almost all financial institutions, especially banks, and insurance firms, have been victimized by financial fraud in one way or another. Traditionally, manual verification, inspection, and other such methods have been employed to identify fraudulent activities, without integrity, high in cost, and time-consuming. Artificial Intelligence and Machine Learning now provide a more efficient means for intelligent detection and prevention of fraudulent transactions, through the analysis of large financial datasets.

The rapid growth in digital banking in Kenya has fairly promoted financial inclusion, but it also accelerated fraudulent activities that threaten financial institutions and their customers. The old rule-based systems for detecting fraud can hardly keep up with new tricks from fraudsters. This study proposes an unsupervised machine learning model for real-time detection of fraudulent digital banking transactions in Kenya. Using unsupervised learning algorithms, the model will detect unique anomalies and suspicious activities that deviate from ordinary customer behavior within the Kenyan digital banking realm. The model will be trained and validated with a real-world dataset of Kenyan digital banking.

Keywords: artificial intelligence; fraud; detection; fraudulent banking operations; machine learning; artificial intelligence.

VT OMNES VNVM SINT

Table of Contents

VT OMNES VNVM SINT

List of Figures

List of Tables

List of Abbreviations

AED-LGB	Autoencoder-Decoder with Light Gradient Boosting Machine
AI	Artificial Intelligence
AML	Anti-Money Laundering
ANN	Artificial Neural Network
API	Application Programming Interface
AUC	Area Under the Receiver Operating Characteristic Curve
CNN	Convolutional Neural Network
DT	Decision Tree
FITEE	*Frontiers of Information Technology & Electronic Engineering*
GA	Genetic Algorithm
GB	Gradient Boosting
GBM	Gradient Boosting Machine
GNB	Gaussian Naive Bayes
IForest	Isolation Forest
IoT	Internet of Things
IT	Information Technology
ITrees	Isolation Trees
KNN	K-Nearest Neighbors
KSQL	Kafka Structured Query Language (Kafka Streams SQL)
LOF	Local Outlier Factor
LR	Logistic Regression
LSTM	Long Short-Term Memory
MAC	Media Access Control
ML	Machine Learning

MMEAs	Multi-Modal Evolutionary Algorithms
MMOP	Multi-Modal Multi-Objective Problem
MOP	Multi-Objective Problem
NACOSTI	National Commission for Science, Technology, and Innovation
NB	Naive Bayes
OC-SVM	One-Class Support Vector Machine
PCA	Principal Component Analysis
RADStack	Real-time Analytics Data Stack
RF	Random Forest
RNN	Recurrent Neural Network
RUS	Random Under-Sampling
SGD	Stochastic Gradient Descent
SMOTE	Synthetic Minority Over-sampling Technique
SOM	Self-Organizing Map
SSD	Structured Systems Development
SSRN	*Social Science Research Network*
SVM	Support Vector Machine
TOS	Transformed Outlier Scores
UTC	Coordinated Universal Time
XGBoost	Extreme Gradient Boosting
XGBOD	Extreme Gradient Boosting Outlier Detection

Definition of Terms

Anomaly Detection	A technique to identify rare events or outliers that deviate significantly from normal patterns. In this study, it refers to flagging transactions with abnormal characteristics indicative of fraud (Steinwart et al., 2005).
Anomaly Score	A numerical value assigned by the Isolation Forest algorithm (0 to 1), quantifying the likelihood of a transaction being fraudulent. Higher scores indicate greater deviation from normal behavior (F. T. Liu et al., 2012)
Event Streaming	Revolves around the unbounded, sequential and real-time flow of data records, called "events," foundational data structures that record any occurrence in the system or environment (Klous, 2010).
Fraud Detection	The process of identifying and preventing deceptive transactions or activities intended for illicit financial gain, using analytical methods to distinguish fraudulent behavior from legitimate operations (Jones, 2017).
Isolation Forest (IForest)	An unsupervised algorithm that isolates anomalies by randomly partitioning data in feature space. It assigns anomaly scores based on the ease of separation, with higher scores indicating greater fraud likelihood (F. T. Liu et al., 2012).
Real-Time Processing	Immediate analysis of data streams i.e transactions as they occur, enabling instant fraud detection and response typically within milliseconds (Collett et al., 2004).
Unsupervised Machine Learning	A type of machine learning where algorithms infer patterns from unlabeled data without predefined outcomes. It identifies hidden structures, such as anomalies or clusters, without human supervision (Abbassi et al., 2024)

Acknowledgement

I am deeply thankful to God for His grace and guidance, which have been my strength throughout my studies, research, and the completion of this report. I would like to express my heartfelt gratitude to my professor, Prof. Ismail Ateya, and my supervisor, Dr. Nelson Ochieng, for their invaluable guidance, timely feedback, and the countless hours they dedicated to helping me refine my work. Their expertise and encouragement have been instrumental in shaping this project. I am grateful to my family for their support and encouragement during this period of my studies, especially my wife and daughter, who encouraged and provided support throughout this journey.

Dedication

This thesis is dedicated to my family, whose unwavering love and support have been the foundation of my academic and personal journey. To my wife, Winny Bett and daughter Natalie Lagat, who have always believed in me, encouraged me to pursue my dreams, and provided me with the resources and guidance to succeed. Your sacrifices and dedication have been a constant inspiration, and I am forever grateful for the opportunities you have given me.

Chapter 1: Introduction

1.1 Background

“Over the past years, a technological revolution has occurred on the Internet that paved the emergence of modern services, especially in e-commerce and money transfer”. E-commerce is one of the many economic domains in information and communications technology that contributed to business improvement, paved the way for managing medium and small companies, reducing costs and saving time, and increasing productivity (Al-Hashedi & Magalingam, 2021). Financial systems face a rapidly evolving threat from various fraud types, including online banking fraud, credit card fraud, fraudulent loan applications, document falsification, phishing scams, and the creation of fraudulent accounts. In a juniper research (Malone, 2023), estimates that Losses from Online Payment Fraud to Exceed $362 Billion Globally between 2023 and 2028, leading to not only direct financial losses but also reputational damage, regulatory scrutiny, increased operational costs, and a decline in customer trust.

As technology continues to reshape financial systems, traditional fraud detection methods are struggling to keep pace. The sheer volume and complexity of modern financial transactions demand innovative, adaptive solutions to stay ahead of increasingly sophisticated fraud tactics. Fraudsters are constantly evolving their strategies, creating a "cat-and-mouse game" that requires equally dynamic and intelligent systems to detect and prevent fraudulent activities in real time. (Kamuangu, 2024). The ongoing digital transformation of financial services, while offering convenience, has also expanded the attack surface for fraudsters, necessitating more sophisticated and adaptable detection mechanisms (Familoni & Shoetan, 2024). According to Jiang & Broby (2021), Credit card and online banking transactions, which are currently most banking system transactions, all present additional vulnerabilities. Innovation in technology has opened new channels that now expose commercial banks in Kenya to cases of financial fraud.

The convergence of technical advancements and employee expertise within banking presents ethical challenges. Individuals with deep system knowledge may be more readily positioned to perpetrate fraud, highlighting the need for robust internal controls (Owiti et al., 2023).

Financial systems face a wide range of fraudulent activities, from online banking fraud and credit card scams to fraudulent loan applications, document falsification, and phishing schemes. These crimes don’t just drain millions of dollars annually from financial institutions, they also erode customer trust and confidence (West & Bhattacharya, 2016). Whether it’s a fake account or a

cleverly disguised phishing email, each type of fraud poses a unique challenge, highlighting the need for robust, adaptive solutions to protect both businesses and their customers.

Financial fraud is increasingly posing a significant challenge for both customers and financial service providers. There have been many cases reported where customers have lost money through online banking and card transactions (Rouhollahi, 2021). Fraudulent activities pose significant threats to business and customers across various sectors, including banking, insurance, e-commerce, and healthcare (Rouhollahi, 2021). Each day, fraud tactics evolve and become increasingly sophisticated leaving the traditional rule-based fraud detection algorithms and systems no chance. Machine learning and artificial intelligence offer promising avenues to augment fraud detection and prevention efforts by enabling the identification of complex patterns and anomalies indicative of fraudulent behavior (Mytnyk et al., 2023)

Financial institutions and the communities they serve would greatly benefit from a comprehensive fraud detection framework that leverages the power of machine learning and event streaming to identify and mitigate fraud in real-time (Dr. Sathisha & Dr. Sowmya, 2023; Palaiokrassas et al., 2023).

1.2 Research Problem

The phenomena of technological breakthroughs have brought a sea change in modern life and promoted a shift from physical services to digital channels. Money transfer and other financial transactions through online platforms are some changes in the banking sector, showing the transition (Neves et al., 2023). However, as financial systems become increasingly digitized, conventional fraud detection methods are proving inadequate in addressing the evolving sophistication of digital fraud. With fewer physical bank branches and a growing reliance on digital banking, financial institutions must adopt more advanced, automated approaches to mitigate these emerging threats (*The Multi-Faceted Threat of Fraud - KPMG Global*, 2022).

According to (Alghofaili et al., 2020), global financial institutions and corporations face significant setbacks as a result of numerous instances of fraud. Despite the development of advanced fraud-prevention tools such as chip-and-PIN verification, 3D Secure for online transactions (Benchaji et al., 2021), and e-banking authentication systems, conventional machine learning techniques remain insufficient for accurately distinguishing between fraudulent and legitimate transactions. This is mainly due to the fact that, conventional machine learning algorithms tend to overlook

changes and trends in consumer purchasing behaviors and cybercriminals develop novel deceptive practices to evade detection, frequently altering their methods (Alghofaili et al., 2020; Zhou et al., 2021). As a result, financial institutions face significant challenges in safeguarding their systems and customers from fraud-related losses.

To address these challenges, this study proposes a self-adaptive fraud detection system powered by AI/ML technologies, designed to dynamically evolve with emerging threats by integrating real-time anomaly detection, behavioral analytics, and automated model retraining. These features enable the system to:

i). Adapt to Changing Fraud Patterns: By continuously analyzing transactional data and consumer behavior, the system can identify and respond to new fraud tactics as they emerge.

ii). Enable Proactive Fraud Prevention: The system's ability to detect anomalies in real time allows for preemptive action, reducing financial losses and protecting customers.

By embedding resilience into transactional workflows, the proposed system not only counteracts financial losses but also fortifies the integrity of digital banking ecosystems. his is particularly critical in emerging markets like Kenya, where the rapid adoption of digital banking necessitates robust, scalable, and adaptive fraud detection frameworks.

1.3 Aim

The objective of this study is to develop an unsupervised machine learning model for digital banking platforms capable of detecting fraudulent transactions in real time.

1.4 Specific Objectives

i). Investigate the challenges and limitations of fraud detection in digital banking transactions.

ii). Evaluate unsupervised machine learning models for fraud detection.

iii). Develop an unsupervised machine learning model for real-time fraud detection and prevention.

iv). Validate the performance of the developed model using appropriate evaluation metrics.

1.5 Research Questions

i. What are the challenges and limitations in real time digital banking fraud detection?

ii. How do different unsupervised machine learning algorithms perform in detecting fraudulent transactions?

iii. How can an unsupervised machine learning model be designed and developed?

iv. How can the developed model be validated?

1.6 Justification

The outcome of this research is significant for several reasons. First, fraud poses a substantial threat to businesses, organizations, and individuals, leading to financial losses and reputational damage. By developing more effective fraud detection systems, the research can contribute to mitigating these risks and improving overall security. Second, as AI and machine learning become increasingly prevalent in various domains, understanding their potential in fraud detection is crucial for staying ahead of sophisticated fraudulent activities. Finally, the research findings can guide policymakers, organizations, and technology developers in making informed decisions regarding the implementation and utilization of AI-based fraud detection systems.

1.7 Scope of the study

The dissertation has the potential to gain a thorough comprehension of AI and ML applications in fraud detection, pinpoint major obstacles, suggest creative solutions, and delineate possible directions for future studies within the specified timeframe. Nevertheless, the creation of practical implementation strategies in real-world scenarios and the thorough validation of proposed methodologies may necessitate extra time and resources that exceed the dissertation's scope.

1.8 Limitations

Studies have focused on mobile and online banking fraud limits its applicability to other financial fraud domains, such as insurance or loan fraud, which may require distinct detection strategies. Reliance on the Kaggle dataset despite its realism introduces geographical bias, as it reflects European transaction patterns rather than Kenya's digital banking ecosystem. Synthetic data, while privacy-compliant, may oversimplify fraud dynamics compared to proprietary banking logs. Additionally, the Isolation Forest model's computational efficiency (0.5ms per transaction) prioritizes real-time processing over explainability, potentially complicating auditability for regulatory compliance. Lastly, the research does not account for adversarial attacks designed to evade ML-based detection. Future work could address these gaps by collaborating with Kenyan

banks for localized data, integrating explainable AI (XAI) modules, and testing hybrid models against adversarial scenarios.

1.8.1 Domain-Specific Limitations and Ethical Considerations

Domain Challenges

While synthetic datasets (e.g., Faker-generated data) mitigate privacy concerns, they oversimplify fraud dynamics. For instance, synthetic transactions lack nuanced behavioral patterns (e.g., multi-step account takeover sequences) observed in real-world Kenyan mobile-money fraud (Owiti et al., 2023).

Ethical Risks

Overly sensitive anomaly thresholds may block legitimate transactions, disproportionately impacting unbanked populations reliant on mobile-money services. Fraudsters may exploit model explainability gaps (e.g., perturbing transaction amounts to evade detection).

Chapter 2: Literature Review

2.1 Introduction

There is often confusion between bank fraud and bank robbery. While bank robbery involves physical violence and direct theft, bank fraud is typically a covert and gradual process that often goes unnoticed until it is too late. Bank fraud usually involves tricking customers into revealing sensitive banking information, which is then exploited to steal funds remotely.

Fraud detection involves measures aimed at preventing the illegal acquisition of money or property through deception. In the banking sector, fraud can take many forms, such as check forgery, the use of stolen credit cards, inflating losses, or fabricating events like accidents to claim payouts. According to the Nilson Report (2024), global card fraud losses were projected to reach a staggering $34 billion by 2024, highlighting the scale and urgency of addressing this issue..

Fraud detection, as explained by Cameron (2023), is all about keeping a close eye on transactions and customer behavior to identify and combat fraudulent activities. It's a key part of any company's strategy to prevent losses and often ties into broader anti-money laundering (AML) compliance efforts. At its core, fraud detection is about stopping people from gaining money or property through deceitful means. In the world of digital banking, fraud detection has been transformed by artificial intelligence (AI). Many businesses now use AI-powered tools to enhance their fraud detection systems. Machine learning (ML) and AI can sift through massive amounts of data in no time, spotting red flags like unauthorized transactions or unusual behavior patterns. Thanks to these technologies, banks are now better equipped than ever to prevent financial fraud and safeguard their customers' hard-earned money (Writer, 2023).

Fraud detection and prevention software comes in both proprietary and open-source options. These tools typically include a range of features designed to help businesses stay ahead of fraudulent activities. Common features include dashboards for easy monitoring, data import/export, and visualization tools for spotting trends. They also often integrate with customer relationship management (CRM) systems and offer calendar management, budgeting, and scheduling capabilities. Additionally, many solutions support multi-user access, password and access management, APIs for seamless integration, and two-factor authentication for enhanced security. Billing and customer database management are also standard features, making these tools a comprehensive solution for tackling fraud (*What Is Fraud Detection and Prevention?*, n.d.).

2.2 Empirical Literature

The increasing prevalence of online financial transactions has made them an attractive target for cybercriminals, necessitating the development of effective fraud detection mechanisms. Research indicates that conventional methods, such as application layer anomaly detection, often struggle to identify fraudulent behaviors due to the sheer volume of internet traffic and the sophisticated nature of cyberattacks. (Arfeen & Khan, 2023) propose a multi-layer machine learning framework that integrates intrusion detection at the network layer with fraud detection at the application layer, addressing the limitations of existing detection systems. This framework leverages advanced machine learning techniques to enhance the detection of anomalies within electronic fund transfer transactions.

Benchaji et al. (2021) explored the effectiveness of machine learning in detecting e-commerce fraud using a recent dataset containing various variables. They tested four key algorithms: Decision Trees (DT), Logistic Regression (LR), Random Forest (RF), and Extreme Gradient Boosting. Among these, Logistic Regression (LR) emerged as the top performer, achieving an impressive accuracy of 0.93 and a precision score of 0.95, outperforming the other models. This highlights LR's potential for identifying fraudulent transactions in e-commerce, though the study also underscores the importance of selecting the right algorithm based on the specific context and dataset characteristics. Sahin & Duman (2011) compared Artificial Neural Networks and Logistic Regression for credit card fraud detection using a real-world dataset. While both models performed similarly on the training data, ANNs demonstrated superior performance on the test data. However, the authors acknowledge that ANNs require extensive training to generate accurate predictions. Consequently, they suggest that ANNs are better suited for classification tasks rather than real-time fraud or anomaly detection, which often require immediate responses to potentially fraudulent activities.

Focusing on the financial sector Bagga et al. (2020) compares the performance of various machine learning algorithms, including logistic regression, random forests, and unsupervised techniques like quadrant discriminative analysis, for credit card fraud detection. The study concludes that unsupervised methods, particularly when combined with pipelining techniques, exhibit promising

results in identifying fraudulent transactions. The authors emphasize the importance of feature engineering and data preprocessing for enhancing the performance of unsupervised models.

Sharma et al. (2021) conducted a comparative analysis of machine learning algorithms, evaluating supervised models such as Random Forests (RF), Logistic Regression (LR), Support Vector Machines (SVM), and Artificial Neural Networks (ANN) on a transactional dataset. Their results showed that ANN outperformed the other models, achieving an F1 score of 0.91. While machine learning algorithms offer significant advantages for detecting bank transaction fraud, they also have notable limitations. One key challenge is their reliance on large volumes of data to achieve accurate results, which can hinder their practical application in fraud detection(Stojanović et al., 2021).

Mutemi & Bacao (2023) proposed an automated fraud detection system using a large transactional database from an online marketplace to identify potential fraud in organized retail banking. Their study evaluated seven supervised machine learning algorithms: Logistic Regression (LR), Decision Trees (DT), Support Vector Machines (SVM), k-Nearest Neighbors (KNN), Random Forests (RF), Gaussian Naive Bayes (GNB), and Gradient Boosting (GB). While GNB achieved the highest recall value of 95.4% among the models, it struggled to identify genuine positive cases, resulting in the lowest accuracy of just 40%.

In a related study, Mytnyk et al. (2023) compared seven machine learning models—RF, KNN, LR, Stochastic Gradient Descent (SGD), DT, Naive Bayes (NB), and SVM—on a transactional dataset. Their findings revealed that LR performed well, achieving an AUC value of 94.6%, while SGD outperformed all models with the highest AUC of 95.4%

Furthermore, Onu et al. (2023) introduced a novel approach for detecting fraud in Ponzi schemes within the Ethereum network. Their study employed three machine learning algorithms—Random Forest (RF), Artificial Neural Networks (ANN), and k-Nearest Neighbors (KNN)—to analyze a dataset of over 20,000 Ethereum interaction channels sourced from Kaggle. After evaluating the performance of each model, Random Forest emerged as the top performer, achieving an accuracy of 94%, an average score of 88.33%, and an overall rating of 96.6%.

Du et al. (2023) proposed the AED-LGB method for detecting suspicious transactions. This approach uses an autoencoder to extract key features from the data, which are then fed into a LightGBM model for classification and prediction. The method was tested on an anonymized dataset with imbalanced transactions. While the SMOTE technique was applied to oversample the

data and improve its quality, the results showed that the AED-LGB-SMOTE combination didn't outperform the standard AED-LGB model. This suggests that the AED-LGB method is better suited for handling the imbalanced data commonly found in financial fraud detection. Additionally, the AED-LGB model, even without data enhancement, outperformed other classifiers like KNN, Random Forest, and LightGBM, highlighting its effectiveness in this domain. On the other hand, Ahmad et al. (Ahmad et al., 2023) tackled the challenge of imbalanced data by using the Random Under-Sampling (RUS) technique. Their goal was to improve accuracy and achieve stronger results by testing various machine learning algorithms. They introduced a method that uses fuzzy C-means clustering to group the dataset and then selects similar fraudulent and legitimate instances based on shared characteristics. The experimental results demonstrated that their approach is effective in handling imbalanced data and improving fraud detection outcomes.

The isolation forest algorithm has gained recognition as one of the most effective methods for detecting anomalies, particularly in the context of fraudulent digital banking transactions. Recent studies, such as that by Palekar et al. (2020), have employed both the Isolation Forest (IForest) and Local Outlier Factor (LOF) algorithms to detect fraudulent credit card activities, yielding promising results. Similarly, Rajeev & Devi (2022), delved into different unsupervised learning techniques for detecting transactional fraud, specifically focusing on IForest and LOF. Their findings revealed a striking difference in performance between the two algorithms. The Isolation Forest model clearly outperformed LOF, boasting a fraud detection rate of around 27%, compared to LOF's modest 0.02%. When it came to accuracy, the Isolation Forest achieved an impressive score of 0.99774, significantly outpacing the Local Outlier Factor. This highlights the Isolation Forest's potential as a more reliable tool for identifying fraudulent transactions.

In addition, (Hajek et al., 2023) proposed a novel XGBoost-based framework for detecting fraud in mobile payment systems, combining both supervised learning and unsupervised outlier detection. The study compares five machine learning approaches—random forest, decision trees, logistic regression, k-nearest neighbors, and autoencoders—using a dataset of over six million transactions to ensure realism. While the hybrid XGBoost-based model (XGBOD) demonstrates enhanced fraud detection accuracy, it also increases execution time, potentially limiting its use in real-time environments with constrained computational resources.

2.3 Theoretical Literature Review

The prevention of financial fraud through effective internal control mechanisms is based on several theoretical frameworks and concepts (Musyoki, 2023). This review aims to explore and analyze these frameworks, providing a deeper understanding of the principles that shape the design and implementation of internal controls focused on fraud prevention. By delving into these foundational theories, we can gain a clearer perspective on how effective internal controls are crafted and applied to protect against fraudulent activities. This examination will highlight the theoretical underpinnings that guide the creation of robust systems to detect, prevent, and mitigate financial fraud.

2.3.1 The Fraud Triangle Theory

The timeless Fraud Triangle, introduced by Cressey (1953), continues to be a fundamental model for understanding the drivers and enablers of fraudulent behavior. At its core, this framework suggests that fraud arises when three critical elements align: *pressure* (the incentive or motivation to commit fraud), *opportunity* (the conditions that make fraud possible), and *rationalization* (the mindset that justifies unethical actions). Over the years, this foundational theory has been built upon and refined, with recent research integrating modern insights and evolving trends in fraud detection, further enriching our understanding of how and why fraud occurs.

2.3.1.1 Pressure

While financial pressures remain a significant driver of fraud Apostolou et al, (2020), recent studies highlight the increasing relevance of non-financial pressures, such as meeting performance targets, career advancement aspirations, or maintaining a certain lifestyle Additionally, the COVID-19 pandemic introduced unique pressures related to economic uncertainty and remote work environments, potentially increasing fraud risk (Bhasin & Gulati, 2021).

2.3.1.2 Opportunity

Technological advancements have significantly altered the fraud landscape, creating new opportunities for perpetrators. The rise of digital payments, cryptocurrency, and online platforms has expanded the attack surface, making it easier for fraudsters to operate across borders and remain anonymous. Furthermore, weaknesses in cybersecurity infrastructure and data protection

measures provide fertile ground for data breaches and identity theft, further enabling fraudulent activities.

2.3.1.3 Rationalization

Modern rationalization techniques often involve justifying actions by claiming a lack of awareness of regulations, shifting blame to others, or minimizing the perceived impact of the fraudulent act (Apostolou et al., 2020). The anonymity afforded by the digital world can further facilitate rationalization, as perpetrators may feel detached from the consequences of their actions (Kaspersky, 2023).

2.3.1.4 Beyond the triangle

While the Fraud Triangle provides a valuable framework, recent literature acknowledges its limitations and proposes complementary theories. The Fraud Diamond, for instance, adds a fourth element: capability, emphasizing the skills and resources required to execute fraud successfully (Wolfe & Hermanson, 2004). Others argue for a more holistic approach, incorporating organizational culture, ethical leadership, and individual moral development as crucial factors influencing fraud risk (Schuchter & Levi, 2015)

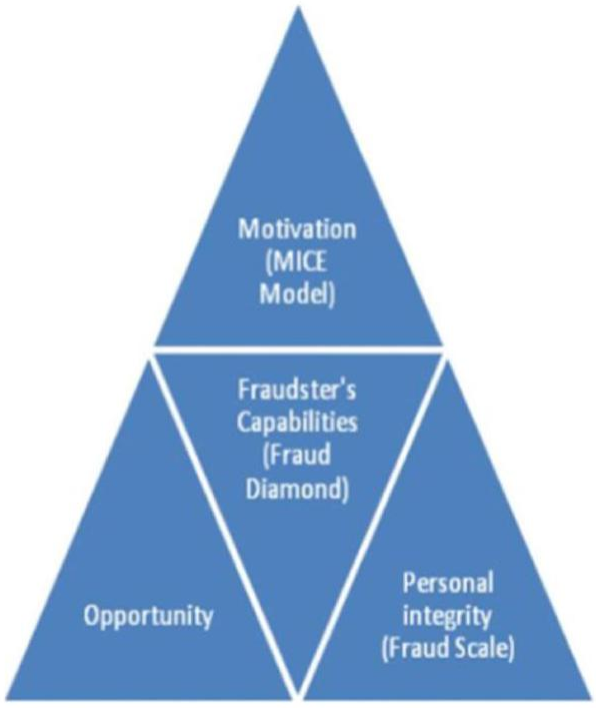

Figure 2.1: Fraud Diamond

The updated fraud triangle offers a more nuanced and comprehensive approach to understanding and tackling fraud in today's ever-changing socio-economic landscape. Its goal is to help organizations and investigators develop better strategies for identifying and minimizing the risks tied to fraudulent behavior.

2.3.2 Agency Theory

The concept of agency theory was initially introduced by Ross (1973) and Mitnick (1973) who described it as a conflict of interest that emerges between principals (owners) and agents (managers). While both perspectives share a similar foundational principle, Mitnick viewed the agency problem as an issue rooted in institutional structure, whereas Ross framed it as a matter of incentives (Mitnick, 2019).

Agency theory is a widely used framework in economics and management that analyzes the intricate relationships between principals (owners or stakeholders) and agents (individuals or entities entrusted with carrying out tasks on behalf of the principals). This theory investigates the challenges and potential conflicts of interest that arise when the goals of principals and agents are not fully aligned. It is particularly relevant in situations where ownership and control are separated, such as in corporations where shareholders delegate decision-making authority to executives (Musyoki, 2023). By applying strategies grounded in agency theory, organizations can improve their ability to detect fraud and foster a culture of trust and accountability.

2.4 Models and Frameworks

2.4.1 Models

Fraud detection within the realm of digital banking has emerged as a vital focus for both research and development, particularly as financial transactions have progressively transitioned to online environments. The importance of fraud detection lies in its role in preserving the integrity of banking systems and safeguarding users against potential financial losses. As machine learning (ML) techniques have advanced, there has been a notable shift towards the integration of data-driven approaches in fraud detection methodologies. Unsupervised learning models are especially noteworthy in this context, as they operate without the need for labeled datasets, which are frequently scarce due to the infrequent nature of fraudulent activities.

2.4.1.1 Extreme Gradient Boosting with Random Under-Sampling

Hajek et al (2023) introduced an approach called Extreme Gradient Boosting combined with Random Under-Sampling, which utilizes the strengths of XGBoost—its supervised learning capabilities and robustness—along with data sampling techniques to tackle the issue of class imbalance often found in mobile payment data. Detecting financial fraud is particularly challenging due to the significant imbalance between legitimate transactions and fraudulent ones (DU et al., 2018). Various strategies have been explored to enhance the performance of supervised learning models in such scenarios. Among these, data-level solutions, as highlighted in research by Pambudi et al. (2019), have proven especially effective. These methods focus on addressing the class imbalance problem before the machine learning models are even trained, leading to better classification outcomes.

XGBoost is a powerful and scalable version of gradient boosted decision trees that constructs models step by step, adding new components to minimize errors progressively (Hajek et al., 2023). It works by iteratively building models that focus on correcting mistakes made in earlier steps, gradually reducing the overall error. To make the method more resilient to noise and less prone to overfitting, gradient boosting was enhanced with random sampling, leading to what's known as stochastic gradient boosting. XGBoost takes this a step further by incorporating additional regularization techniques, which help prevent overfitting and improve model performance (Chen & Guestrin, 2016).

$$\text{obj}^{(t)} = \sum_{i=1}^{n} (y_i - \left(\hat{y}_i^{(t-1)} + f_t(x_i)\right))^2 + \sum_{t=1}^{T} \Omega(f_t),$$

Equation 2.1; XGBoost Equation

2.4.1.2 Multimodal Multi objective Evolutionary Algorithm

Liu et al (2019) conceptualized the selection of features as a multi-objective problem (MOP) and successfully identifies a small set of features that exhibit high classification accuracy. Conventional MOPs tend to neglect the diversity present in the resolution space, prioritizing the attainment of an optimal solution. To mitigate this challenge, the authors introduce a multimodal multi-objective problem (MMOP) architecture that aims to uncover a strong Pareto front in the target space, as well as several analogous optimal Pareto solutions within the feature space.

Additionally, they propose a competition-driven process designed to enhance current multi-modal evolutionary algorithms (MMEAs), thereby increasing the variety of solutions and aiding in the discovery of the necessary Pareto fronts. The experimental results demonstrate that this approach not only improves classification accuracy but also provides more comparable feature subsets, making it particularly useful for unbalanced classification problems such as credit card fraud detection.

2.4.1.3 Extreme Gradient Boosting Outlier Detection Model

The XGBOD approach, articulated by (Zhou et al., 2018),is a semi-supervised ensemble algorithm that integrates multiple unsupervised outlier detection techniques alongside an XGBoost classifier. The process starts by using unsupervised methods to create transformed outlier scores (TOS), which serve as new data representations. Next, a feature selection technique is applied to reduce the TOS feature space, keeping only the most significant scores. This step combines the outlier score matrix with the original features, creating an enriched feature space. Within this improved space, the XGBoost classifier is then used to produce final outlier scores for each mobile payment transaction. The key advantage of this approach is its high predictive accuracy, which stems from its ability to resist overfitting and effectively manage imbalanced data. In the XGBOD-based fraud detection framework, multiple unsupervised outlier detection methods are used to generate the TOS features. To strike a balance between diversity and accuracy, the balance selection algorithm Zhou et al. (2018) is employed.

$$\Psi(TOS_i) = \frac{AUC_i}{\sum_{i,j=1}^{k} \mid \rho(TOS_i, TOS_j) \mid},$$

Equation 2.2; Extreme Gradient Boosting Outlier Detection

2.4.2 Frameworks

In the realm of fraud detection, unsupervised machine learning frameworks are particularly valuable for identifying anomalies in datasets without labeled instances. These frameworks

provide the necessary tools for developing, training, and deploying complex models in an efficient and scalable manner.

2.4.2.1 PyTorch

PyTorch, developed by Facebook's AI Research lab, is another popular deep learning framework, particularly favoured by researchers for its flexibility and ease of experimentation. The dynamic computation graph utilized by PyTorch allows for real-time modifications to the network architecture, which enhances its intuitiveness relative to TensorFlow's static graph approach. This dynamic capability is particularly beneficial for research and experimental purposes, where models often require swift iterations and adjustments (Sunaryono et al., 2019). Furthermore, PyTorch's ease of use in data loading and model training positions it as an excellent choice for real-time fraud detection tasks that involve large datasets.

2.4.2.2 Scikit-learn

Scikit-learn serves as an essential framework for implementing machine learning algorithms due to its user-friendly interface and comprehensive library of tools. Its modular design facilitates the integration of various preprocessing techniques, model selection processes, and evaluation metrics, making it suitable for both novice and experienced practitioners.

2.4.2.3 TensorFlow

TensorFlow, developed by Google, is one of the most widely used open-source machine learning frameworks. It is highly regarded for its flexibility, scalability, and ability to handle large datasets, making it a popular choice for building deep learning models, including facial recognition systems. TensorFlow allows developers to create complex neural networks, such as Convolutional Neural Networks (CNNs), which can be used for fraud detection. In a 2023 study by Kendeya et al., researchers introduced and tested a hybrid CNN-SVM method for detecting credit card fraud, using a real-world transactional dataset. The results showed that the CNN-SVM model performed exceptionally well, achieving a precision of 90.50%, an accuracy of 91.08%, and an F1-score of 90.41% (Kendeya et al., 2023). These metrics highlight the model's effectiveness in accurately identifying fraudulent transactions.

2.5 Architectures and Designs

The rapid expansion of digital payment platforms has significantly increased the need for robust fraud detection mechanisms, especially in the context of digital banking transactions. This section explores the architecture, design, and performance of adaptive fraud detection systems, emphasizing their ability to learn and adapt in real-time from digital banking transaction data.

2.5.1 Standalone Architecture

Traditional fraud detection systems often encounter challenges such as high false positive rates, difficulty in adapting to evolving fraud patterns, and reliance on labeled data. To address these limitations, standalone architectures leveraging unsupervised learning techniques have been proposed.

One notable architecture is the Anomaly Detection Engine, which combines unsupervised learning algorithms such as the One-Class Support Vector Machine (OC-SVM) and Isolation Forest to detect anomalies in financial transactions. This approach has shown promising results in identifying previously unknown fraud patterns without requiring labeled data (Kulatilleke, 2022). These architectures emphasize the development of robust features that are resistant to model drift and concept drift, common challenges in fraud detection. For instance, deep reinforcement learning has been utilized to learn optimal fraud detection policies, enabling the system to adapt its behavior automatically based on environmental feedback.

Another innovative direction involves the use of event streaming platforms like Apache Kafka to facilitate real-time fraud detection (Vimal et al., 2021). The integration of event streaming platforms like Apache Kafka has emerged as a transformative approach to real-time fraud detection in digital banking. Unlike traditional batch-processing methods that introduce delays, Kafka enables low-latency, high-throughput transaction monitoring, allowing fraudulent activities to be identified and mitigated before completion. By leveraging its distributed and fault-tolerant architecture, Kafka efficiently processes millions of transactions per second while seamlessly integrating with machine learning models such as Isolation Forest and Autoencoders for anomaly detection.

2.5.2 Hybrid Architecture

In addition to the standalone models, researchers have also explored hybrid approaches that combine multiple techniques to leverage their respective strengths and address the limitations of

individual methods. These hybrid architectures often involve a combination of clustering, anomaly detection, and graph-based methods, along with feature engineering and ensemble learning, to achieve more robust and accurate fraud detection. Abbassi et al (2024) introduced an architectural framework designed to improve the detection and monitoring of highly complex digital transaction fraud. This framework leverages big data engines, deep learning techniques, and unsupervised learning methods, such as autoencoders and extended isolation forests. The proposed architecture enhances the ability to identify and manage intricate fraud scenarios in digital transactions. Figure 2.2 below illustrates the architecture outlined in their study.

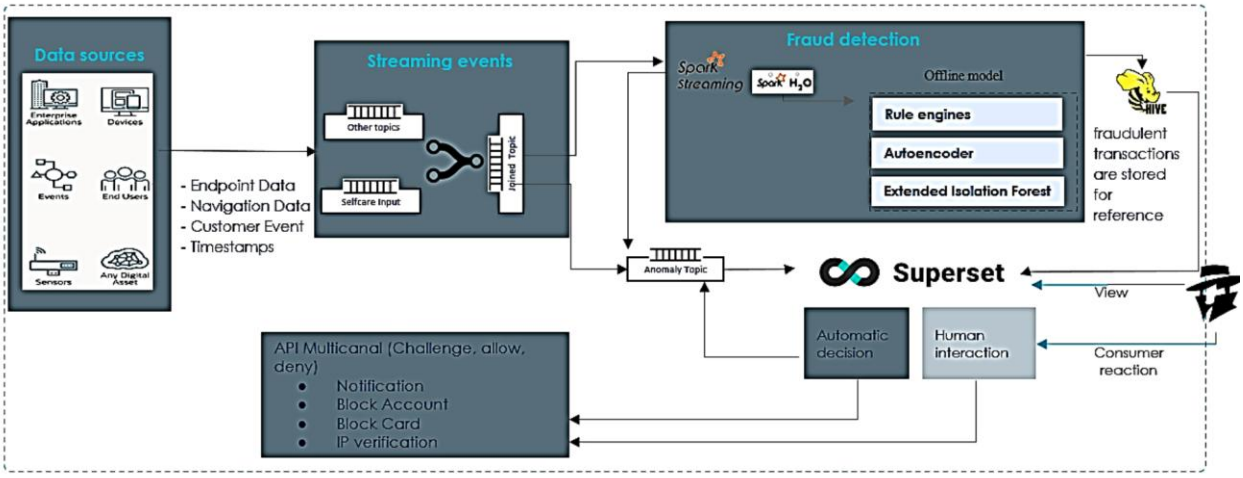

Figure 2.2: Hybrid Architecture

One of the main advantages of hybrid architecture is their flexibility. By distributing some of the processing tasks to the client devices, the system can handle more clients without overloading the central server. This also allows for more efficient use of resources, as the client devices can perform simple tasks, while the server handles more complex operations. Additionally, hybrid architecture can function even if the network connection between the client and server is temporarily disrupted, as the client devices can continue to process data locally.

2.5.3 Streaming Architecture

Streaming data processing is often discussed in relation to big data analytics and has been utilized in numerous applications, including the monitoring of systems, the advancement of smart cities, service evaluation, and marketing initiatives (De Paepe et al., 2021). Lambda and Kappa architectures are widely recognized in this field. The Lambda architecture is highly effective in scenarios that demand both real-time data processing and historical analysis, such as traffic

monitoring Ta-Shma et al. (2018) or situations where data corrections may be needed after initial processing(Yang, 2017). Conversely, the Kappa architecture refines the Lambda approach by removing the batch processing layer, resulting in a more streamlined design. This simplification enhances code reuse, as the same data flow is utilized for both updating existing analytics and computing new ones.

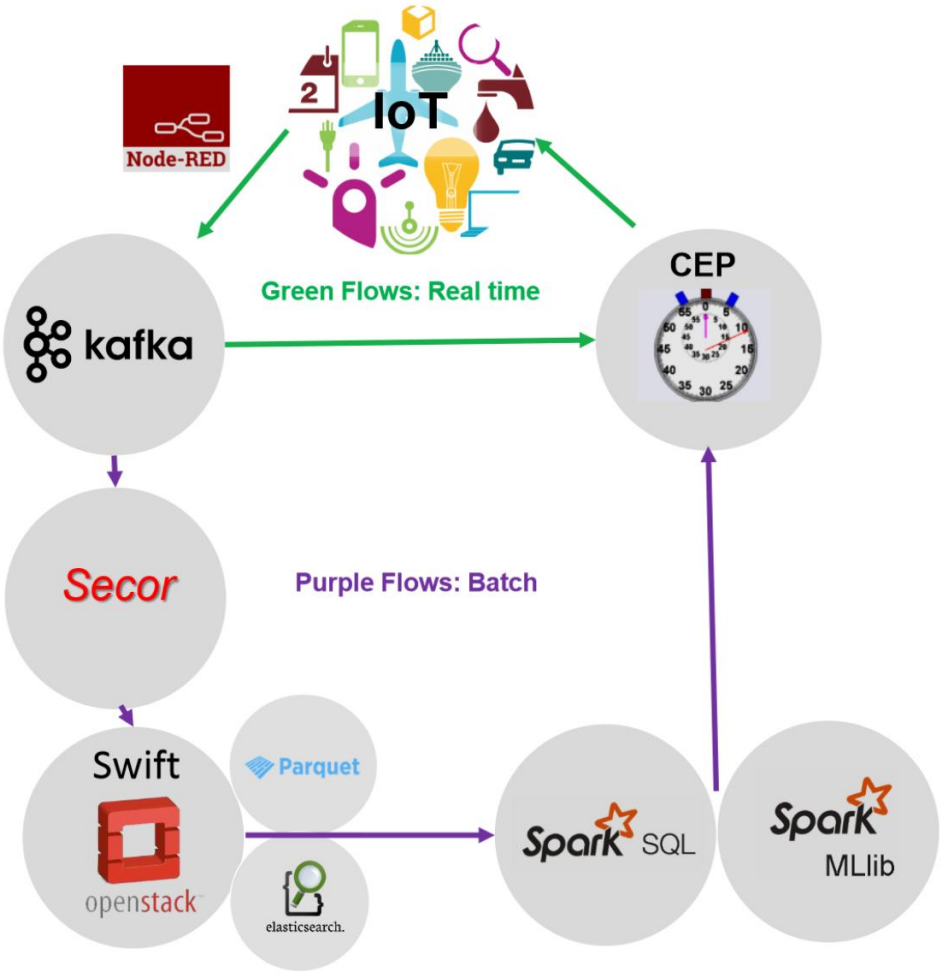

Figure 2.3: Streaming Architecture

2.6 Algorithms

2.7 AE-EIF algorithm

One example of a hybrid approach is the Autoencoder-Extended Isolation Forest algorithm, which integrates deep learning and ensemble learning techniques. The algorithm begins by employing an

autoencoder to learn a compressed representation of the transaction data, effectively capturing the underlying patterns. Following this, an extended version of the Isolation Forest algorithm is applied to these learned representations to detect anomalies (Abbassi et al., 2024). Their approach leverages hybrid models that combine the strengths of autoencoder-based deep learning and extended isolation forest techniques to enhance fraud detection modeling. This model is specifically designed for real-time applications in digital banking environments. However, its practical implementation faces challenges, particularly concerning data privacy issues. Additionally, the computational cost of the proposed framework was not evaluated, raising concerns about the model's overall efficiency and scalability in real-world scenarios.

2.7.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning method that aims to find a maximum margin hyperplane to classify input training data into two categories. It can classify new data points using a labeled training set for each class (HaratiNik et al., 2012). Rajak & Mathai, (2015) proposed a hybrid approach combining SVM with the Danger Theory fusion for fraud detection. Their experimental results demonstrated that this method outperformed existing approaches in terms of time complexity and F-measure, highlighting its effectiveness in detecting fraudulent activities. Kendeya et al. (2023) proposed an architecture that incorporates SVM and CNN into their model for credit card fraud detection in which they achieved 91.08% accuracy though with an imbalanced dataset. One of the main strengths of SVM is its ability to perform well in environments with limited labelled data, which is often a challenge in educational settings. However, SVM’s performance depends on the correct selection of hyperparameters, and improper tuning can lead to overfitting or poor generalization to new data (Kendeya et al., 2023).

2.7.2 Genetic Algorithm

Genetic algorithms, a type of evolutionary algorithm, are designed to iteratively improve solutions over time. Since their inception, they have found success across a wide range of fields—from astronomy to sports, and from optimization challenges to computer science applications. In the realm of data mining, they have proven particularly useful for tasks like variable selection, often working in tandem with other data mining techniques to enhance their effectiveness (Vats, 2013). In a study by Ileberi et al (2022) introduces a novel approach to credit card fraud detection by leveraging a Genetic Algorithm for feature selection in conjunction with five distinct machine

learning classifiers: Random Forest, Decision Tree, Artificial Neural Network, Naive Bayes, and Logistic Regression. The GA employed the RF classifier as part of its fitness function to guide the search for optimal feature subsets within the European cardholder's credit card transactions dataset. This process yielded five high-performing feature vectors.

Experimental results demonstrated the efficacy of the proposed GA-based feature selection method. Notably, the GA-RF model, utilizing the fifth feature vector (v5), achieved an exceptional accuracy of 99.98%. Furthermore, other classifiers, such as the GA-DT, also exhibited remarkable performance, attaining an accuracy of 99.92%. These findings underscore the potential of integrating GA-based feature selection with diverse classifiers to enhance fraud detection accuracy.

2.7.3 Clustering Based Methods

Clustering, an unsupervised learning technique that groups similar data points into clusters (Ahmed et al., 2016), has proven highly effective in financial fraud detection. For instance, Glancy and Yadav (2011) demonstrated its potential by combining hierarchical clustering with text mining to create a fraud detection model for financial transactions. Expanding on this, they also introduced an innovative approach that integrates text dimension reduction—using Singular Value Decomposition—with document clustering to identify fraudulent activity. Their method employs a dual Growing Hierarchical Self-Organizing Map technique, which challenges the traditional assumption that non-fraudulent data lies at the center of the data space. This highlights the adaptability and power of clustering techniques in uncovering complex patterns of financial fraud.

Addressing the limitations of traditional Self-Organizing Maps (SOMs) in handling uncertain cluster boundaries, Majidi (2023) proposed a hybrid model that integrates K-means clustering with SOM. In this approach, SOM is utilized to extract central representatives of data patterns, and K-means is subsequently applied to refine the cluster assignments. This hybrid model effectively reduces ambiguity at cluster boundaries and improves the accuracy and interpretability of the clustering results, particularly in complex datasets.

Table 2.1: Summary of literary works on fraud detection and machine learning

Study	Model or Framework	Drawbacks
Abbassi et al. (2024)	Autoencoder-Extended Isolation Forest	Real-world implementation challenges related to data privacy; computing cost not assessed.
HaratiNik et al. (2012); Rajak & Mathai (2015)	Support Vector Machine (SVM)	Performance depends on hyperparameter tuning; improper tuning can lead to overfitting or poor generalization.
Kendeya et al. (2023)	SVM and CNN for credit card fraud detection	Achieved accuracy with an imbalanced dataset.
Ileberi et al. (2022)	Genetic Algorithm for feature selection	Requires integration with multiple classifiers; results may vary based on dataset characteristics.
Ahmed et al. (2016); Glancy & Yadav (2011)	Clustering methods (hierarchical clustering, text mining)	Traditional SOMs struggle with uncertain cluster boundaries.
Majidi (2023)	Hybrid model of K-means and SOM	Complexity in model integration; may require further refinement for ambiguous datasets.
Hajek et al. (2023)	Extreme Gradient Boosting with Random Under-Sampling	Class imbalance in mobile payment data can still pose challenges; effectiveness depends on proper implementation of sampling methods.
Liu et al. (2019)	Multimodal Multi-objective Evolutionary Algorithm	Complexity in managing multiple objectives; may require significant computational resources to achieve optimal solutions.

Zhou et al. (2018)	Extreme Gradient Boosting Outlier Detection Model (XGBOD)	Performance relies on the selection of unsupervised methods; balancing diversity and accuracy can be challenging.
Sunaryono et al. (2019)	PyTorch	Requires familiarity with dynamic computation graphs; may have a steeper learning curve for those used to static frameworks.
Scikit-learn	Scikit-learn Framework	Limited to traditional machine learning algorithms; may not be as effective for deep learning applications compared to other frameworks.
Kendeya et al. (2023)	Hybrid CNN-SVM Technique	Performance may vary based on dataset characteristics; it requires careful tuning of both CNN and SVM parameters for optimal results.

2.8 Existing Gaps

A comprehensive review of current transactional fraud detection research reveals several significant limitations and unresolved challenges:

2.8.1 Critical Analysis of Supervised vs. Unsupervised Learning in Fraud Detection

While supervised learning methods like Logistic Regression (LR) and Random Forests (RF) dominate fraud detection research due to their high accuracy in labeled datasets (Hajek et al., 2023), their reliance on historical fraud patterns limits their ability to detect novel or evolving fraud tactics. For instance, Zhou et al. (2021) demonstrated that supervised models trained on European credit card data failed to generalize to emerging fraud schemes in mobile-money ecosystems, where labeled fraud cases are scarce and underreported (Owiti et al., 2023).

Unsupervised methods, such as Isolation Forest (IForest), bypass the need for labeled data by identifying deviations from normal transaction behavior. This is particularly advantageous in

Kenya's digital banking landscape, where only 0.17% of transactions are flagged as fraudulent (Kaggle dataset), and fraudsters rapidly adapt tactics (e.g., "airtime-flushing" scams targeting prepaid mobile credits). However, the trade-off lies in interpretability: while supervised models provide feature importance scores (e.g., SHAP values), IForest's anomaly scores lack actionable insights, complicating compliance with Central Bank of Kenya's audit requirements (Banking Fraud Act, 2022).

2.8.1.1 Scalability and Real-Time Deployment Challenges

Real-time fraud detection systems demand low-latency processing and scalability to handle high transaction volumes (e.g., M-Pesa's 12,000 transactions/second). Isolation Forest's linear time complexity O(n)$O(n)$ makes it computationally efficient compared to density-based methods like LOF (O(n2)$O(n2)$) (Chabchoub et al., 2022). However, Abbassi et al. (2024) noted that even linear algorithms struggle with throughput exceeding 50,000 transactions/sec on single-node systems, necessitating distributed frameworks like Apache Spark.

The key challenge in financial fraud detection is *real-time performance*. Banks and mobile-money platforms process thousands of transactions per second, so a fraud system must scale horizontally and respond within milliseconds or seconds. Modern solutions use distributed stream-processing (Apache Kafka, Flink, etc.) to ingest data and score each transaction in real time (Tambi, 2022). As one architectural survey notes, high-speed pipelines capture transaction streams into Kafka/Flink for "minimal latency" before applying ML inference ((Tambi, 2022). Low latency is critical: customers expect instantaneous payments, and detection systems must flag or block fraud within this tight window. For example, Redis Enterprise (an industry platform) advertises sub-millisecond latency for transaction scoring in live environments, highlighting that even small delays can harm user experience.

2.8.2 Predominance of Supervised Learning and its Drawbacks:

Existing literature by Ahmed et al. (2016) indicates a heavy reliance on supervised learning algorithms for fraud detection. This preference stems from the relative ease of implementation compared to unsupervised methods and the scarcity of publicly available labeled datasets. However, this dependence on supervised learning presents notable drawbacks:

i) **Performance Degradation with Unlabeled Data:** Supervised models, when applied to unlabeled datasets a common scenario in real-world settings experience a significant reduction in both detection accuracy and computational efficiency.

ii) **Challenges in Evaluating Effectiveness:** Data privacy concerns pose a significant obstacle to rigorously evaluating the effectiveness of deployed fraud detection models. Access to sensitive transactional data for analysis is often restricted, hindering accurate performance assessments.

2.8.2.1 Limitations of Traditional Machine Learning Models in Detecting Novel Fraud:

An analysis of existing research reveals that traditional ML-based fraud detection strategies struggle to identify previously unknown fraudulent actors or activities. This limitation arises from two key factors:

i) **Data Imbalance and Bias:** The infrequent nature of fraudulent activities leads to imbalanced datasets, where legitimate transactions significantly outnumber fraudulent ones. Consequently, risk prediction models trained predominantly on labeled, approved transactions exhibit a bias towards legitimate behavior and may fail to generalize to unseen fraudulent patterns.

ii) **Neglecting Denied Transactions:** Current approaches often overlook the valuable information present in denied transactions. While denied transactions may not represent actual fraud, they often contain risk indicators that could enhance fraud detection models.

2.8.2.2 Computational Costs of Complex Models:

While sophisticated models like Neural Networks and Decision Trees offer potential for accurate fraud detection, their computational demands pose a practical challenge. Achieving timely and accurate results with these models often necessitates significant computational resources, potentially limiting their scalability and feasibility in certain operational contexts.

2.9 Conceptual Model

To address the challenges of digital banking fraud detection, I propose a conceptual model for an unsupervised learning-based fraud detection system that leverages event streaming. The capacity of machine learning to discern previously unknown fraudulent patterns by analyzing historical data underscores its significance in fraud detection. However, given the limitations of current

approaches, there is a pressing need for a pragmatic fraud detection methodology that effectively addresses these shortcomings (Abbassi et al., 2024).

The conceptual model will harness the power of big data analytics to enhance its capacity to address increasingly complex digital operation fraud scenarios. By leveraging advanced analytical techniques, the model aims to improve detection accuracy, adaptability, and responsiveness in identifying and mitigating sophisticated fraudulent activities in digital environments. This approach ensures a more robust and proactive defense against evolving fraud challenges. The proposed fraud detection model employs a real-time anomaly detection approach leveraging customers' historical interactions with the bank's digital banking ecosystem. This level of analysis aims to identify anomalous transactions by analyzing real-time customer data through a pre-trained Isolation Forest algorithm.

The model assigns an anomaly score to each transaction, effectively predicting the likelihood of fraudulent activity. Transactions exceeding a predefined threshold are flagged as potentially fraudulent and forwarded for human review and the accounts blocked from transacting. This human-in-the-loop approach allows for expert validation or rejection of flagged transactions, ensuring accuracy and minimizing false positives.

Upon confirmation of fraudulent activity, the system enables immediate action through various communication channels. Transaction monitoring agents can initiate corrective measures and alert account holders about potential fraud through mobile application notifications, email, or SMS messages. This multi-channel notification system ensures timely communication and empowers customers to take necessary precautions. The event streaming workflow is shown in figure 4 below.

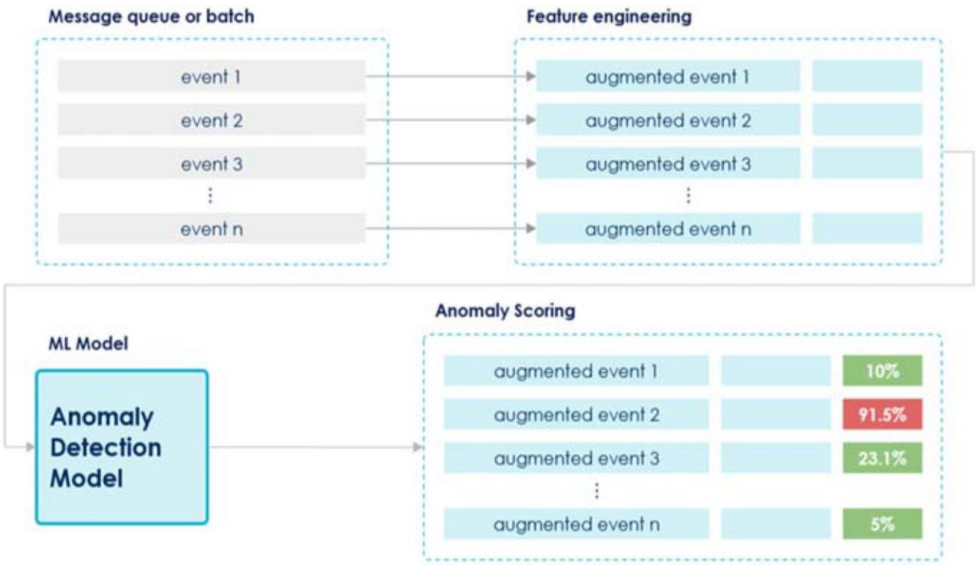

Figure 2.4: Fraud Detection workflow

The isolation forest workflow will be as shown in figure 5.

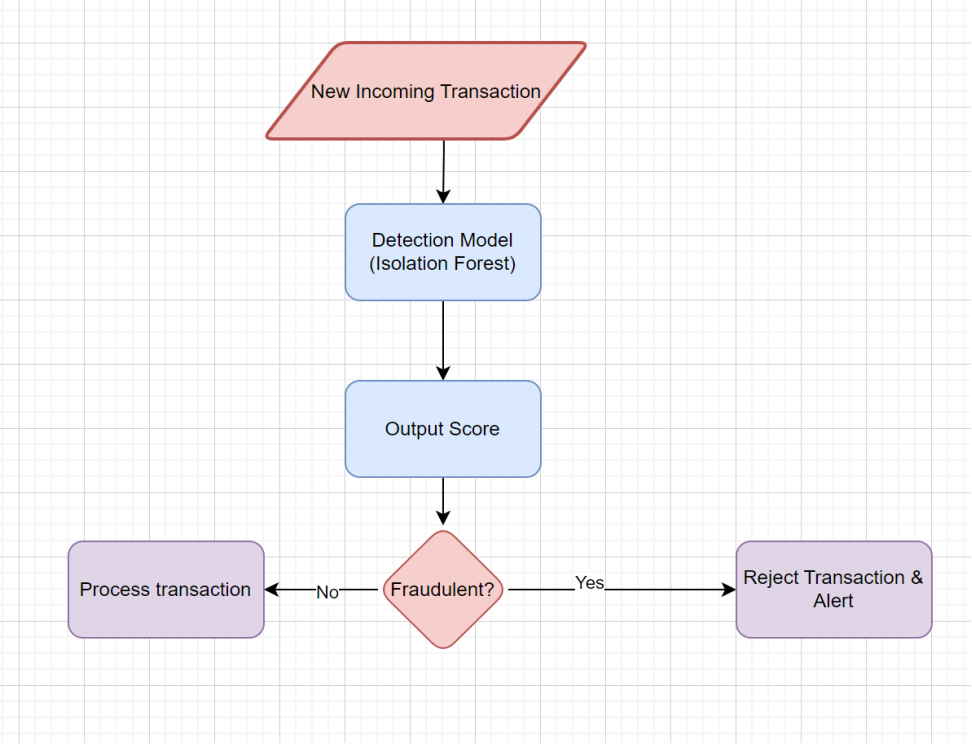

Figure 2.5: Isolation Forest detection flow

2.9.1 Model Architecture

The proposed model is built on an event-driven architecture leveraging real-time anomaly detection through the Isolation Forest algorithm. Below is an expanded explanation of how data flows through the system and the architectural components involved:

2.9.1.1 Data Ingestion Layer

This layer gathers transactions in real-time from a variety of sources, including online banking systems, mobile apps, and ATM interactions. These transactions are then streamed into Kafka, a real-time data pipeline, from diverse channels such as e-commerce platforms, digital banking applications, and other payment systems. This ensures a continuous and seamless flow of data for further processing and analysis.

2.9.1.2 Feature Engineering & Preprocessing

Incoming transactions are enriched with additional contextual features, such as historical transaction patterns, geolocation data, and device fingerprinting. These features are then fed into the anomaly detection model.

2.9.1.3 Anomaly Detection Model (Isolation Forest)

A pre-trained Isolation Forest model, powered by H2O's Sparkling Water, evaluates each transaction and assigns a fraud likelihood score. This model calculates an anomaly score for every transaction by measuring how much it deviates from typical historical behavior. Sparkling Water seamlessly integrates with Spark, enabling efficient training and deployment of the Isolation Forest model for fraud detection. Any transaction that scores above a predetermined threshold is flagged as potentially fraudulent for further investigation.

2.9.1.4 Storage and Alerting

Flagged transactions are stored in PostgreSQL for further analysis. Alerts are generated for human review via email, SMS, and a dashboard. The fraud detection model continuously learns from new fraudulent patterns, improving its accuracy over time.

Chapter 3: Research Methodology

3.1 Introduction

This chapter outlines the research design and philosophy adopted in this study to develop an unsupervised machine learning model for real-time digital banking fraud detection. The research design provides a framework for collecting and analyzing data, while philosophy underpins the approach to knowledge generation and justification. This chapter also discusses the data source, model development, system development, dissemination of results, and ethical considerations that are essential for ensuring the validity, reliability, and integrity of the study.

3.2 Research Design and Philosophy

This study adopts a developmental research design (Reigeluth & Carr-Chellman, 2009), focusing on the iterative creation and evaluation of an unsupervised machine learning model for real-time fraud detection in digital banking. This design is well-suited to the study's primary objective: developing a functional and adaptive solution to a practical problem rather than merely describing or explaining existing phenomena (Richey et al., 2004). The iterative nature of developmental research allows for continuous refinement of the model through design, development, testing, and evaluation, ensuring its effectiveness in real-world applications.

The research philosophy aligns with pragmatism (Morgan, 2013) , which emphasizes practical solutions to real-world problems and prioritizes actionable outcomes. Pragmatism justifies the use of a mixed-methods approach, combining quantitative techniques (e.g., model performance metrics) with qualitative insights (e.g., expert feedback) to comprehensively evaluate the model's strengths and limitations. This approach is relevant in fraud detection, where the dynamic nature of fraudulent activities requires adaptive and context-sensitive solutions. By focusing on practicality and flexibility, pragmatism ensures that the developed model can evolve to address emerging fraud patterns and maintain its effectiveness over time. For example, pragmatism informs the choice of Isolation Forest as the primary algorithm due to its efficiency and scalability, which are critical for real-time applications in digital banking.

3.3 Data Source

This research employs two datasets to train and evaluate the unsupervised machine learning model for real-time digital banking fraud detection: a simulated dataset and a real-world dataset from

Kaggle. This dual approach allows for a more robust evaluation of the model's performance, bridging the gap between controlled experimentation and real-world scenarios.

3.3.1 Simulated dataset

To address the sensitivity of financial data and privacy concerns, a synthetic dataset will be generated using Python's Faker library. This dataset mimics realistic transaction patterns and includes features critical for fraud detection, as shown in table 3.1 below.

Table 3.1: Features of the simulated dataset

Feature	Description
userId	A unique identifier assigned to each customer in the channels. This ID remains consistent across all transactions and events.
account	The total number of accounts associated with a customer within the channels (e.g., savings, checking, loan accounts).
eventType	The category of action or incident recorded in the audit trail (e.g., login attempt, transaction, password change, failed authentication).
eventPayload	A structured data field containing additional event-related properties, such as transaction amount, location, or user-agent details.
eventDescription	A human-readable summary providing context for the recorded event (e.g., "User initiated a fund transfer of Kes500").
deviceId	The unique identifier (e.g., MAC address or device fingerprint) of the device used to perform the action.
ipAddress	The public or private IP address from which the event originated, useful for tracking geographic location and security analysis.
timestamp	A precise timestamp (in UTC format) indicating when the event was recorded, aiding in event sequencing and anomaly detection

This dataset allows for controlled experimentation while maintaining data privacy

3.3.2 Kaggle Dataset

In addition to the simulated data, this research utilizes a publicly available dataset of credit card transactions from Kaggle (*Credit Card Fraud Detection*, n.d.). This dataset provides real-world

transaction data, offering a valuable opportunity to evaluate the model's performance on actual fraud cases. The dataset contains transactions made by credit cards in September 2013 by European cardholders and includes:

- 28 anonymized features (V1-V28) resulting from a Principal Component Analysis (PCA) transformation. While the meaning of these features is not directly interpretable, they capture important information about the transactions.
- Time: The number of seconds elapsed between each transaction and the first transaction in the dataset.
- Amount: The transaction amount.
- Class: A binary label indicating whether the transaction is fraudulent (1) or legitimate (0).

The dataset comprises 284,807 transactions, of which 492 are fraudulent. While the dataset is imbalanced (with a low percentage of fraudulent transactions), this reflects the real-world distribution of fraud and provides a realistic challenge for the model. The use of this dataset allows for evaluating the model's ability to detect fraud in a realistic setting, even with imbalanced data.

3.4 Model Development

The model development process comprises the following stages:

3.4.1 Data Preprocessing

The raw transaction data is preprocessed to prepare it for model training. This includes:

- **Data Cleaning**: Handling missing values and inconsistencies in the data.
- **Feature Scaling**: Scaling numerical features to a standard range to improve model performance. Techniques like standardization or min-max scaling will be used.
- **Feature Engineering**: A crucial aspect of this research is the feature engineering process, which aims to extract meaningful behavioral features from the raw event data. The goal is to capture user behavior patterns over different time windows and identify the most relevant features for fraud detection. These features are derived from the raw event data, which includes attributes userId, account, eventType, eventPayload, eventDescription, deviceId, ipAddress, and timestamp. The feature engineering process focuses on aggregating user activities over time to create user profiles that reflect their typical behavior. These profiles are then used as input to the unsupervised learning model.

3.4.1.1 Exclusion and Inclusion criteria

The inclusion criteria ensure that the dataset comprises digital banking transactions, including mobile and online banking activities, with relevant attributes such as transaction type, timestamp, and amount. Transactions must be complete, representative of real-world scenarios, and span a defined timeframe to capture evolving fraud patterns. Both fraudulent and legitimate transactions are included to enhance model training, with fraudulent cases either verified (for real-world data) or realistically simulated (for synthetic data).

Exclusion criteria remove irrelevant, duplicate, or incomplete transactions to maintain data integrity. ATM withdrawals, in-branch transactions, and system-generated transfers are excluded as they involve different fraud dynamics. Transactions with missing key attributes or anomalies (e.g., negative amounts) are filtered out. Additionally, data outside the defined timeframe is omitted to ensure relevance to current fraud trends.

3.4.2 Algorithm Selection

This research proposes the use of the Isolation Forest algorithm for unsupervised fraud detection, given its suitability for identifying anomalies in high-dimensional datasets. The Isolation Forest is an unsupervised machine learning technique designed for outlier detection. It operates by constructing a collection of randomly generated isolation trees (iTrees) to identify anomalies. Each tree in the model evaluates whether a specific data point is an outlier. If most of the isolation trees classify a data point as an outlier, it is likely considered an anomaly. This approach isolates anomalies by recursively partitioning the data, effectively distinguishing irregularities from normal data points (Chabchoub et al., 2022). It operates based on two key characteristics of anomalies:

- **Rarity**: Anomalies are uncommon and typically constitute only a small fraction of the overall dataset.
- **Distinct Behavior**: They exhibit patterns that differ significantly from the rest of the data.

Isolation Forest was chosen over alternatives (e.g., Autoencoders, LOF) due to:

- **Efficiency**: Linear time complexity (n)$O(n)$), suitable for real-time processing.
- **Robustness**: Handles high-dimensional data without hyperparameter tuning (Chabchoub et al., 2022).

- **Interpretability**: Anomaly scores provide actionable insights for fraud analysts.

3.4.2.1 Isolation Forest Hyperparameter Tuning and Validation

Parameter Tuning

The Isolation Forest algorithm will be configured using the following hyperparameters:

- n_estimators (number of trees): Tested values (50, 100, 200).
- max_samples (subsample size): Evaluated (0.5, 0.75, 1.0) of the training data.
- contamination (expected outlier proportion): Set initially to 0.0017 and varied over 0.001, 0.002, 0.005 to test robustness.

A grid search over all combinations was conducted using bootstrap cross-validation: for each fold, I, sampled 80% of the data, trained the model, scored the remaining 20%, and aggregated anomaly scores. The combination with the highest average Area Under the Precision-Recall Curve (AUPRC) computed on a small, labeled holdout was chosen as default (n_estimators=100, max_samples=0.75, contamination=0.002).

Unsupervised Validation Metrics

As Isolation Forest is inherently unsupervised, AUPRC was complemented with the measures below:

- **Score Distribution Analysis**: We plot the empirical distribution of anomaly scores and verify a clear separation between bulk and tail and compute the Kullback–Leibler divergence between score distributions at different contamination settings to ensure stability.
- **Repeatability:** measure the standard deviation of anomaly scores for the same 1 000 transactions across 10 training runs (with different random seeds). A low $\sigma < 0.02$ indicates score consistency.

Sensitivity Analysis

To assess robustness, we vary each hyperparameter one at a time:

Table 3.2: Isolation Forest Parameter tuning

Hyperparameter	Range Tested	Key Observation
contamination	0.001 – 0.005 (step 0.001)	AUPRC peaks near 0.002; setting >0.003 yields diminishing returns and more false positives.
n_estimators	50, 100, 200	Marginal AUPRC gains beyond 100; training time doubles at 200.
max_samples	0.5, 0.75, 1.0	Best trade-off at 0.75; lower sample size slightly reduces detection power.

3.4.3 Model Training:

The chosen algorithm was trained on the preprocessed transaction data. During the training phase, the Isolation Forest algorithm constructs a collection of randomly generated isolation trees (iTrees). In the subsequent scoring phase, the algorithm calculates an anomaly score for each data point in the dataset. These scores reflect the likelihood of a data point being an outlier, with higher scores indicating a greater probability of being an anomaly. This two-step process enables the model to effectively identify and quantify irregularities within the data.

- **Training**: 80% of data used to build isolation trees.
- **Validation**: 20% holdout set evaluated using AUC-ROC and precision-recall curves.
- **Threshold Tuning**: Optimized to minimize false positives while maintaining a 95% fraud detection rate.

3.4.4 Fraud Detection:

The trained model will analyze new transactions in real-time, assigning anomaly scores to flag potential fraud. Transactions exceeding a predefined threshold are flagged for further review, enabling proactive fraud prevention.

3.5 System Development

This research adopted the Agile methodology for the development of the unsupervised fraud detection model. Agile is an iterative, flexible, and customer-focused approach to project

management and software development that will enable the team to deliver value incrementally, allowing for quick adjustments based on real-time feedback (Guerrero-Ulloa et al., 2023).

The system development approach to be applied in this research will be Agile Methodology. This methodology will allow for continuous improvements to the different modules of the system, based on the progress of the research and the discovery of new technologies to enhance the functionality of the proposed tool. Most importantly, it will enable the researcher to refine the system requirements incrementally throughout the development process (Lu & DeClue, 2011).

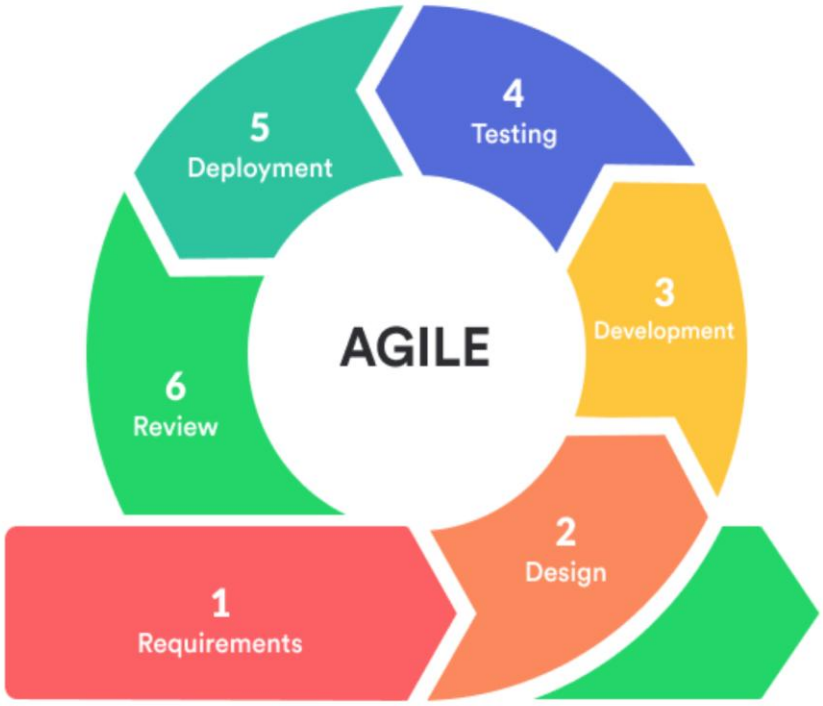

Figure 3.1: Agile Methodology Representation

3.5.1 Architecture

The proposed architecture, designed to process digital transactions as efficiently as possible in real-time, is illustrated in Figure 3.2: Model Architecture. This architecture is divided into three primary components.

Sources: This layer collects data from various inputs, including customer engagement service centers, demographic data from the bank's customer master, known device-to-customer mappings,

and online or mobile banking activity logs. These sources provide the necessary raw data for fraud detection and prevention.

Prevention Layer: This layer processes incoming data streams using Kafka topics. It utilizes Streaming KSQL and the Kafka API to merge data streams and generate a Results Topic, which is further analyzed. The goal is to detect anomalies early and flag suspicious transactions before they proceed further.

Detection Layer: This layer applies advanced analytics using Spark Streaming and H2O.ai, storing relevant findings in PostgreSQL. The data is processed through machine learning models to identify fraud patterns. The detected anomalies are then fed into a Fraud Detection Studio, which interacts with React.js and Node.js for visualization and decision-making. Automated actions, such as notifications, account or card blocking, and IP verification, are triggered through an API Multicanal system, ensuring real-time fraud prevention.

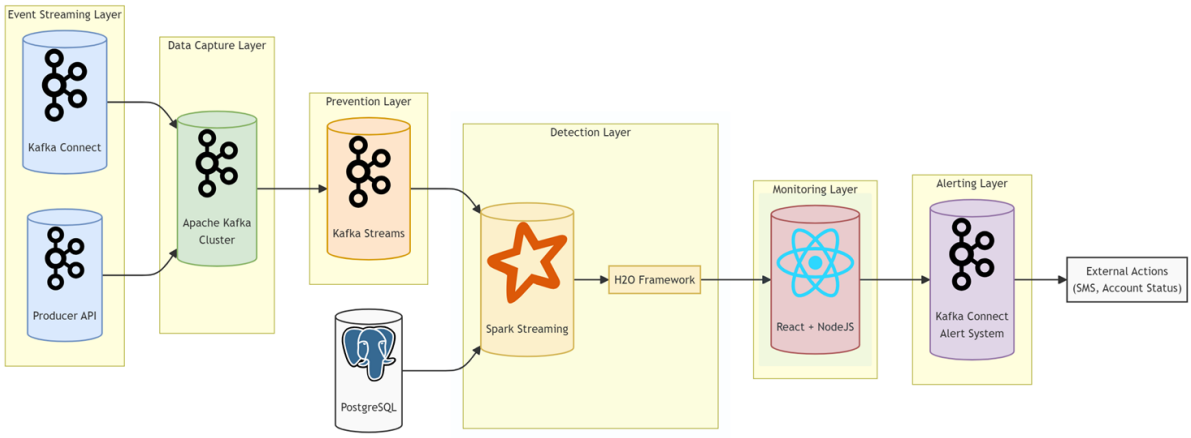

Figure 3.2: Model Architecture

3.5.1.1 Deployment

The developed model will be integrated into a simulated real-time environment. This will involve:

i. **Real-time Data Stream Simulation:** A mechanism to simulate a continuous stream of transaction data will be implemented.

ii. **Model Integration:** The trained machine learning model will be integrated with the data stream. This might involve creating an API or using a message queue system.

iii. **Alert Generation:** When the model detects a potentially fraudulent transaction, an alert will be generated. The alert will include relevant information about the transaction, such as the transaction amount, time, and location.

iv. **Visualization and Monitoring:** A dashboard will be developed to visualize the model's performance and display real-time alerts. This will allow for monitoring the system and evaluating its effectiveness.

3.5.1.2 Review

In this stage, we will review the results from the first round of testing the model and integrate them into the requirements for the next iteration.

3.5.2 System Design

System design refers to architecture, components, modules, interfaces and data for a system to satisfy specified requirements (Waldo, 2006). This study will employ various UML diagrams, including context diagrams, data flow diagrams (levels 1 and 2), data models, database schemas, and wireframes to illustrate the graphical user interface of the tool. These visual representations will effectively convey the design and functionality of the proposed tool in accordance with the SSD methodology.

3.6 Target population and Sampling

The target population for this study comprises financial transactions executed on digital banking platforms, specifically focusing on mobile and online banking activities. Given the sensitivity of financial data and the challenges of accessing real-world datasets, this research utilizes two primary data sources:

i). **Simulated Dataset**: A synthetic dataset generated to mimic realistic transaction patterns.

ii). **Kaggle Dataset**: A publicly available dataset of credit card transactions(*Credit Card Fraud Detection*, n.d.)

3.6.1 Sampling Strategy

For the simulated dataset, a stratified sampling approach is employed to ensure balanced representation of various transaction types i.e. login attempts, fund transfers, and failed authentications for the simulated dataset. This guarantees diversity in the data, enabling the model to generalize across different scenarios.

For the Kaggle dataset, the sampling strategy addresses the inherent class imbalance (492 fraudulent transactions out of 284,807 total transactions). To mitigate this imbalance, Synthetic Minority Over-sampling Technique (SMOTE) is applied to the fraudulent class, ensuring that the model is adequately trained to detect rare anomalies without overfitting the majority class.

To calculate the minimum sample size required, we use the Cochran's Formula, which is widely used for large populations:

$$n_0 = \frac{Z^2 \cdot p \cdot (1-p)}{E^2}$$

Equation 3.1: Cochran's Formula

Where:

- n0 = required sample size
- Z= Z-score (1.96 for 95% confidence level)
- p = estimated proportion of fraud cases (0.172% = 0.00172)
- e = margin of error (typically 5% or 0.05)

$$n_0 = \frac{(1.96)^2 \times (0.00172) \times (1-0.00172)}{(0.05)^2}$$

$$n_0 = \frac{3.8416 \times 0.001716}{0.0025}$$

$$n_0 = \frac{0.00659}{0.0025} \approx 2{,}636$$

Thus, at 95% confidence and 5% margin of error, a sample of 2,636 fraud cases is required for meaningful statistical inference. Since fraud cases are rare in real-world datasets (typically less than 1% of transactions), a dataset of at least 500,000 total transactions is recommended to ensure that enough fraud cases are captured.

3.6.2 Scientific Justification

i). **Simulated Dataset**:

The simulated dataset consists of 1,000,000 transactions, designed to reflect realistic transaction distributions and fraud prevalence rates. This volume is justified based on the following considerations.

Justification of the sample size

- **Industry Benchmarking**: Real-world fraud detection systems typically analyze millions of transactions. However, for experimental research, a dataset in the range of 100,000–500,000 transactions is considered sufficient for model generalization. The larger size of 1,000,000 transactions ensures robustness and scalability.
- **Data Representativeness**: A stratified sampling approach ensures that all major transaction types (e.g., mobile money transfers, bill payments, cash withdrawals) are proportionally represented, avoiding bias toward specific categories.
- **Computational Feasibility**: While large, the dataset size is manageable with distributed computing tools like Apache Spark and H2O.ai, ensuring efficient model training and validation without excessive computational overhead.

ii). **Kaggle Dataset**:

The Kaggle Credit Card Fraud Detection dataset consists of 284,807 transactions, with an inherent class imbalance (0.172% fraudulent transactions).

3.6.3 Sampling Process

i). **Simulated Dataset**:

Generate 1,000,000 synthetic transactions using Python's Faker library provided by PaySim mobile money simulator, stratifying the dataset by transaction type (e.g., 40% fund transfers, 30% login attempts, 20% failed authentications, and 10% others), and inject 1,000 synthetic fraud cases (1% of the dataset) to mimic real-world fraud prevalence.

ii). **Kaggle Dataset**:

The Kaggle dataset consists of 6354407 transactions, with fraud making up only 8213fraud cases. To ensure better fraud detection, SMOTE (Synthetic Minority Oversampling Technique) is used to oversample fraud cases by 5%.

- Initial Fraud Cases = **8213**
- Required Fraud Cases (5%) = **5% of 50,000**

50,000×0.05=2,500

New Dataset Size = 50,000 Transactions (with fraud cases increased to 2,500)

3.7 Dissemination of Research Results

The results of this study will be disseminated through multiple channels to maximize impact:

i). **Scholarly Publications**: Submitted to high-impact journals specializing in fintech and machine learning to ensure rigorous peer review and visibility.

ii). **Conference Presentations**: Shared at leading forums such as the *IEEE International Conference on Data Science and Advanced Analytics (DSAA)* and *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, with a focus on sessions addressing fraud detection in emerging markets.

iii). **Open-Source Contribution**:

a. **Code Repository**: Publish on GitHub under an MIT license, including detailed documentation, Jupyter notebooks for reproducibility, and Docker containers for seamless deployment.

iv). **Stakeholder Engagement**:

a. **Policy Briefs**: Summaries tailored for policymakers to guide regulatory frameworks for AI-driven fraud detection in East Africa.

This multi-pronged approach ensures the research advances academic discourse, supports industry innovation, and informs public policy, aligning with the study's goal of fostering trust in Kenya's digital financial ecosystem.

3.8 Ethical Considerations and False Positives

All data used in the study was anonymized to protect user identities and comply with Kenya's Data Protection Act (n.d.). Sensitive information (e.g., names, phone numbers) will be excluded from the dataset. The data utilized in this research will be secondary, derived from the Kaggle (*Credit Card Fraud Detection*, n.d.) which has already been anonymized and is PCA compliant, Paysim Mobile money simulator dataset, which lacks unique identifiers that could connect any individual to specific transactions. Consequently, any personal information had been removed from the dataset, thereby eliminating the risk of data privacy infringements.

False positives can block legitimate transactions, harming customer trust. We set the anomaly-score threshold to achieve a false-positive rate <2% in our holdout, and we log all flagged cases for human review within 30 minutes to minimize disruption. Privacy & Compliance: All features conform to the Kenya Data Protection Act (2019). Raw PII fields are tokenized in transit, and no model output exposes identifying information. Audit trails record every scoring decision for compliance audits.

Additionally, the research will also undergo ethical review by Strathmore University's Ethical Review Board to ensure that it meets all ethical requirements for research involving human subjects, along with a Research Permit from the National Commission for Science and Technology and Innovation (NACOSTI).

Chapter 4: System Design

4.1 Introduction

This chapter expounds on the analysis and design of the unsupervised machine learning model for digital banking, by incorporating the various requirements that were identified after successful data collection and analysis.

This section provides detailed specifications of the system, including stakeholder definitions, component descriptions, data model specifications, and process model descriptions. It also includes UML diagrams for visual system representation

4.2 Data Preprocessing

Data preprocessing is a crucial step before training a model. In this step, I prepare the data for training by checking for missing values, handling duplicates, and identifying any outliers. If any columns are categorical, we perform encoding since many machine learning models cannot handle categorical values directly. The goal of this analysis was to better understand the simulated dataset and the European union credit card dataset, pre-process it and create a model for prediction.

4.2.1 Data analysis

The goal of this analysis was to better understand the synthetic mobile money transaction dataset, pre-process it and create a model for prediction. The data analysis method applied was exploratory data analysis (EDA. The Paysim mobile money Dataset for fraud detection, comprised of 6362620 observations and 11 columns. The figure below shows the dataset

```
#Loading the dataset
file_path = 'paysim_dataset.csv'
data = pd.read_csv(file_path)

#display the dataset
data.head()
```

[3] ✓ 26.3s

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0

Figure 4.1: Paysim dataset

The simulated data had the attributes below.

Table 4.1: Attributes of the simulated dataset

Column Name	Data Type	Description
user_id	Integer	Unique identifier for the user
account	String	Bank account number in SAFARICOM money format
event_type	String	Type of banking activity (login, transaction, etc.)
event_payload	JSON Object	Detailed transaction/action metadata
event_description	String	Human-readable summary of the event
device_id	String	Unique device identifier
ip_address	String	Originating IP address (Kenyan format)
timestamp	Integer	Unix epoch timestamp in milliseconds

From the Kaggle dataset, a check for duplicates was done and below is the result.

Figure 4.2 Handling duplicates in the data

4.2.2 Handling Imbalanced Data

Unrelated transaction types were filtered out and only what was relevant was maintained. The dataset is highly imbalanced with 99.83% normal transactions and 0.17% fraudulent transactions. This imbalance could affect model performance. To prevent this, data under sampling technique was used to prevent bias towards the majority class under sample dataset.

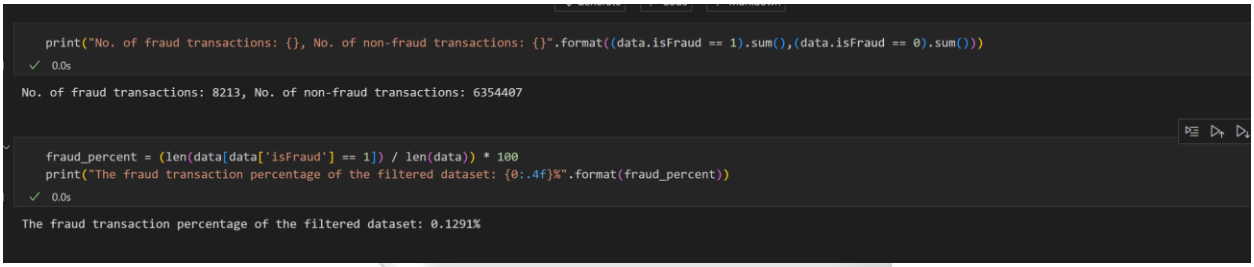

Figure 4.3: Handling imbalances in the data

4.3 Requirements Analysis

This research aims at developing an unsupervised machine learning model for digital banking g fraud detection. Based on this objective, this section outlines the functional, non-functional, and technical requirements for developing an unsupervised machine learning model for real-time digital banking fraud detection.

4.3.1 Functional requirements

These are the key model functionalities that must be met and they include:

i). The system must analyze transactions in real time to flag anomalies as they occur.

ii). Automatically generate alerts for flagged transactions and escalate high-risk cases to fraud analysts.

iii). Build and update user profiles based on transaction history, device usage, and geolocation.

iv). Seamlessly integrate with existing digital banking and core banking systems.

4.3.2 Non-Functional Requirements

These describe the constraints under which the tool must work within, hence the following considerations.

(i) Usability

An intuitive and responsive dashboard is provided, built with React.js and Node.js, to facilitate easy navigation and operation for fraud analysts. The interface supports role-based access control, customizable reporting, and real-time alert management, ensuring a user-friendly experience tailored to the needs of both technical and non-technical stakeholders.

(ii) Accuracy

The solution is built to achieve a fraud detection rate of at least 95% while maintaining a false positive rate of 2% or less. It incorporates AI/ML-based risk models that provide real-time fraud scoring, ensuring that suspicious transactions are flagged promptly, and that the system continuously refines its detection accuracy through ongoing learning mechanisms.

(iii) Data Security

The data being processed by the system is confidential and should be treated as such hence the need for user rights and roles on data management to be assigned as per a company's data policies.

4.4 System Process

The fraud detection system employs a multifaceted and structured approach to detect suspicious activities as it occurs in "real-time." At the outset, the system regularly collects transaction data such as transaction amounts, location, time of transaction and user behavior patterns, from the banking database. Once collected to obtain transaction activity, the data in the database begins a "pre-processing" effort where the data is updated, cleaned, structured, transformed and normalized for any inconsistencies. After pre-processing has completed, the "feature extraction" phase is next where key features such as transactional frequency, historical activity patterns, and customer profiles are extracted to help disclose anomalies. The fraud detection algorithm then utilizes unsupervised machine learning algorithms "theoretically" to identify anomalies and include features such as clustering and anomaly detection to flag questionable transactions, which enables

a user in the banking security team (to be identified in this process workflow) to either allow the transaction, limit approval to further review, or decline action altogether. If the user determines some action should any flagged transactions, then the user can either block the transaction, temporarily stop, or send it for manual approval. It is also important to note the continuous learning that will occur through the detection of fraud over time. Each fraud incident will provide additional information to enhance the model and its usage. Over and "over" this phase will help the model adapt over time to have improved accuracy.

4.5 Data Flow Diagrams

Data flow diagrams (DFDs) illustrate how data moves within the fraud detection system, detailing the interactions between users, databases, and processing units

4.5.1 Context Diagram

The Context Diagram shows a top-level overview of the fraud detection system and the various external entities such as customers, databases for banks, and fraud analysts that are engaged. The customers start the transaction, and the system evaluates the transaction including risk factor assessment. If warranted, the system also communicates with fraud analysts regarding flagged transactions for subsequent review and reaction.

VT OMNES VNVM SINT

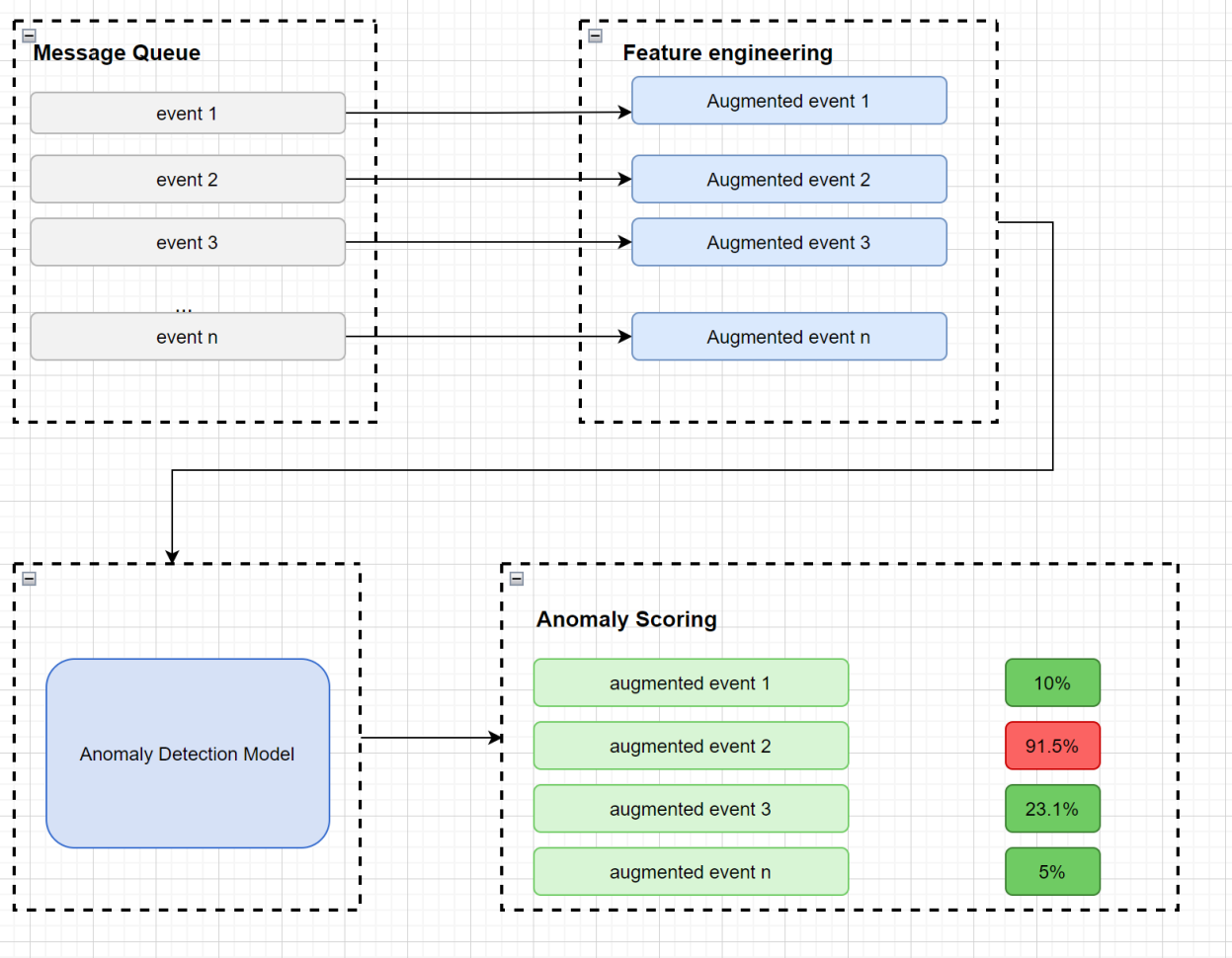

Figure 4.4: Fraud detection workflow context diagram

4.5.2 Data Flow Diagram Level 1

The Level 1 Data Flow Diagram divides the system down into important subsystems. The process starts with customers making transactions that flow into a data preprocessing module to clean and prepare the data for analysis. The fraud detection engine analyzes transaction patterns and anomalies while the alert management system generates alerts on suspicious transactions. If the transaction is flagged, fraud analysts will review the transaction and take proper action based on the risk level.

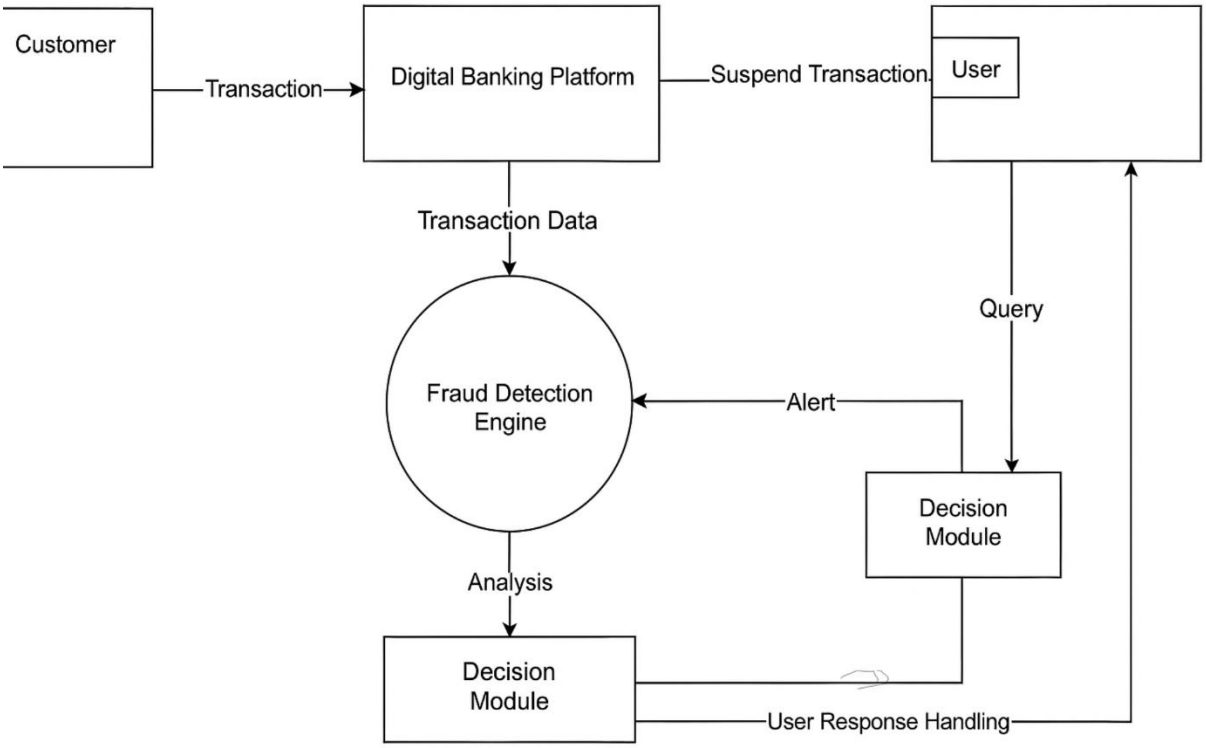

Figure 4.5: Level 1 Data flow diagram

4.5.3 Data Flow Diagram Level 2

The Level 2 Data Flow Diagram breaks the fraud detection engine into more granular components. The unsupervised model component identifies patterns in a set of historical data, a data subset where the component identifies what is to be anticipated and highlights anomalous behavior in the expected activity. Any anomalies are assessed by the rule validation module, which provides known rules of activity and format to assure consistent processing of each data component activity. The machine learning model build, update approach continues to assure the earlier identified transactions modeled after a fraud, are still accurately captured.

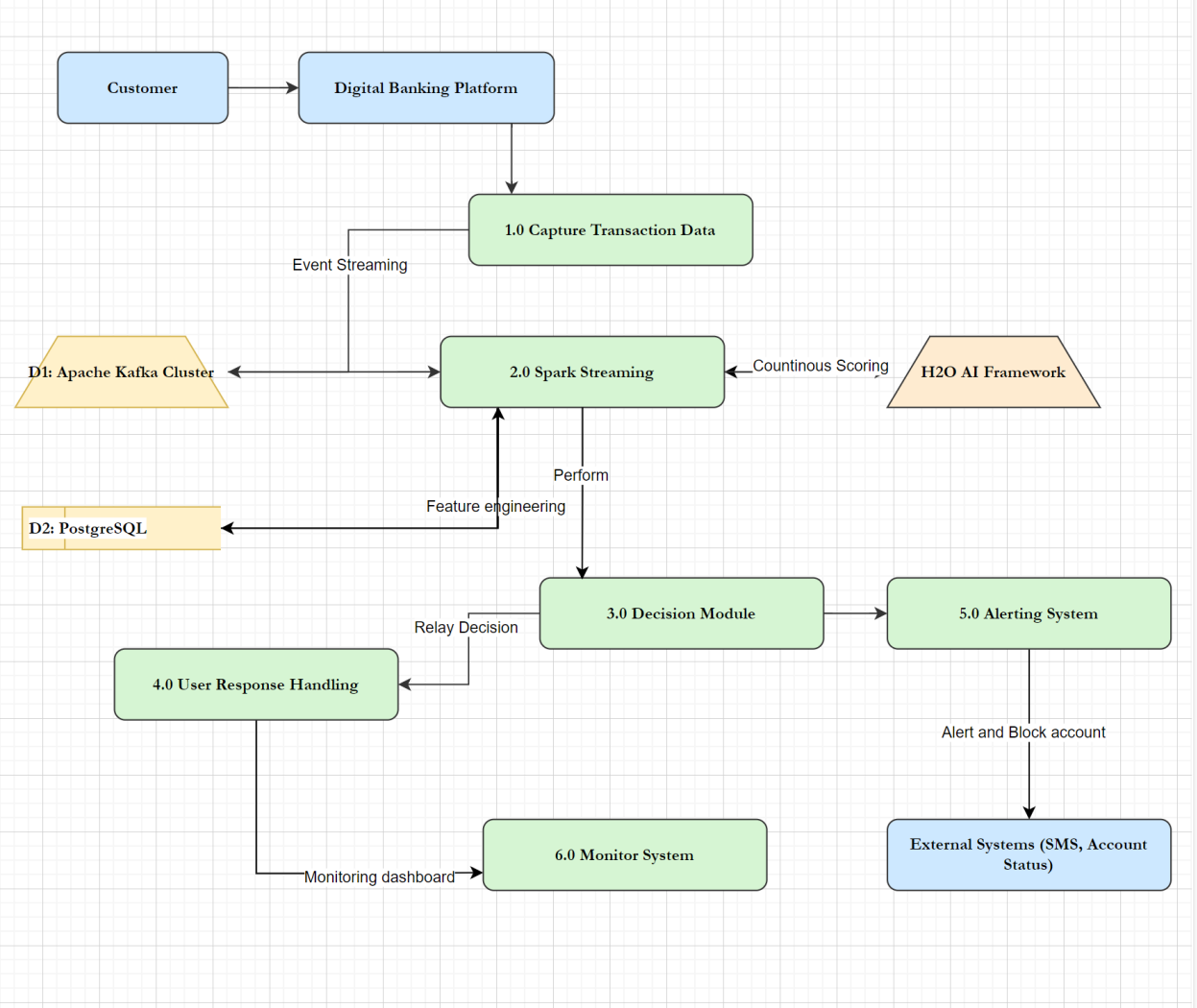

Figure 4.6: Level 2 dataflow diagram

4.6 Data Model

The data model depicts the logical representation of entities and relationships within the fraud detection system. The most important entities involved in the data model consist of transactions, customers, fraud alerts, and analysts (users). These entities and their relationships build the foundational structure supporting fraud detection and response capabilities.

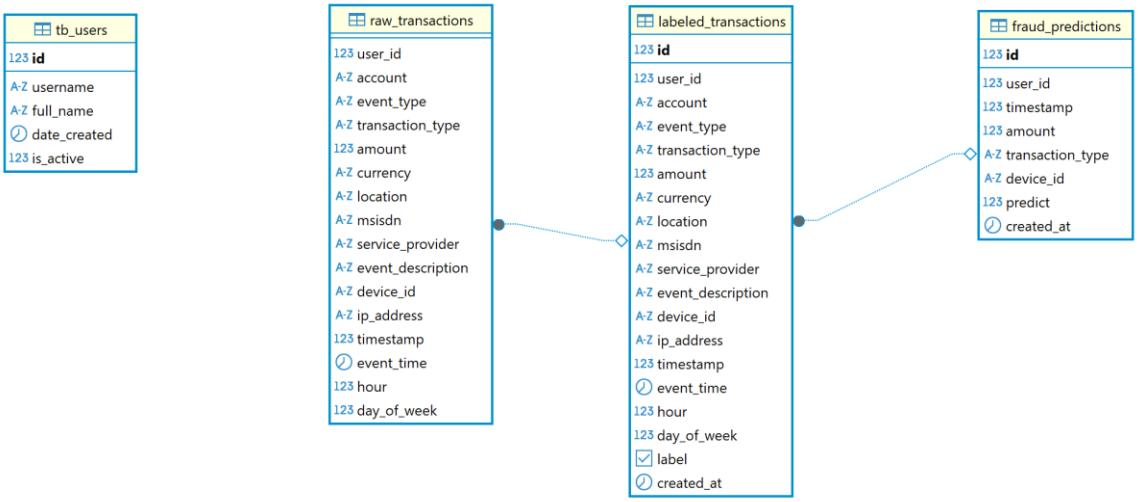

Figure 4.7: Database schema

4.7 Database Schema

The database scheme organizes data storage utilized by the unsupervised model for efficient access and processing. Transactions are captured in two stages: Raw Transactions record initial event details;, user_id, amount, location, device_id, event_time, while Labeled Transactions table inherit these fields and add fraud verdicts. The fraud predictions table serves as a record of alerts including user id, transaction type, amount, device, prediction, status. Lastly, the users table maintains information about security analysts' staff who review transactions flagged as fraudulent and includes username, name, role, and actions taken. In summary, this schema organizes the data systematically and facilitates the efficient and effective processing of fraud detection tasks.

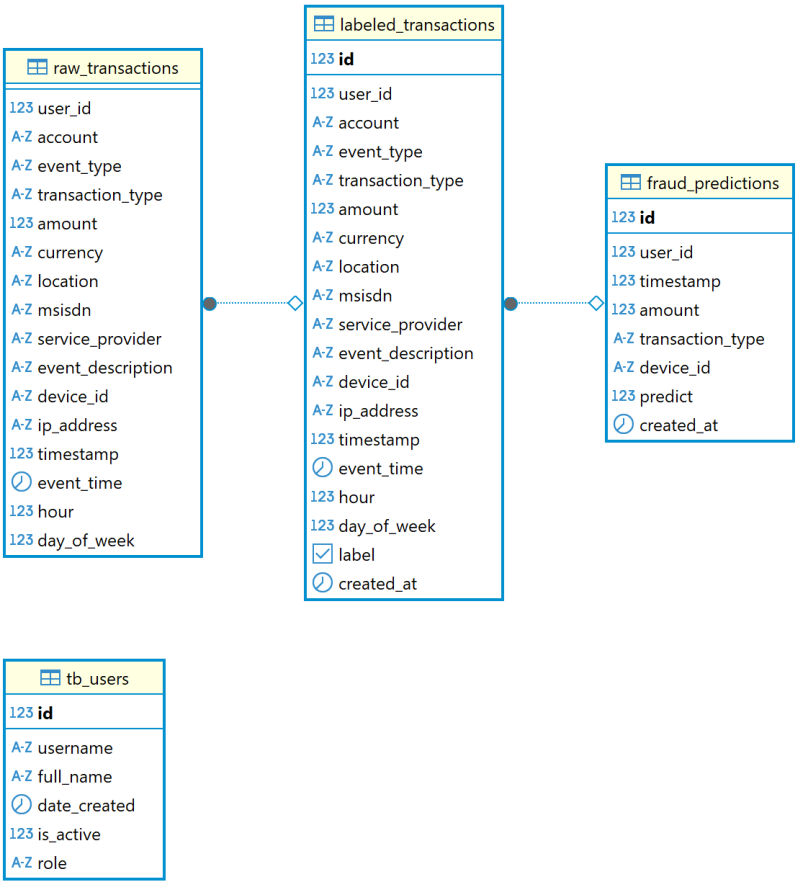

Figure 4.8: Data model

Chapter 5: System Development and Testing

5.1 Introduction

This chapter discusses the building and validation of the unsupervised machine learning model including data preprocessing, model training, and evaluation. Unlabeled data is used for the prediction of fraudulent activities as class & non class. Data is test-sets to determine their consistency and effectiveness.

5.1.1 Importing Transactional Data Source

The first step in the model development process involves acquiring and importing transactional data through simulated event streams from the digital banking platform and Paysim Data from Kaggle. The dataset contains transactions, including attributes such as transaction amount, time, location, and user details. This data serves as the foundation for training the fraud detection model. The figure below shows the paysim dataset

```
#Loading the dataset
file_path = 'paysim_dataset.csv'
data = pd.read_csv(file_path)

#display the dataset
print(data.shape)
print(data.info())
data.head()
```

[4] ✓ 24.1s

```
(6362620, 11)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
 #   Column          Dtype
---  ------          -----
 0   step            int64
 1   type            object
 2   amount          float64
 3   nameOrig        object
 4   oldbalanceOrg   float64
 5   newbalanceOrig  float64
 6   nameDest        object
 7   oldbalanceDest  float64
 8   newbalanceDest  float64
 9   isFraud         int64
 10  isFlaggedFraud  int64
dtypes: float64(5), int64(3), object(3)
memory usage: 534.0+ MB
None
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0

Figure 5.1: Paysim dataset

Properties | Data | Diagram

raw_transactions | Enter a SQL expression to filter results (use Ctrl+Space)

	user_id	account	event_type	transaction_type	amount	currency	location	msisdn	service_provider	event_description
1	565	SAF75336905	transaction	PESALINK_DEPOSIT	79,176	KES	KE	[NULL]	[NULL]	PESALINK_DEPOSIT transactio
2	248	SAF32286788	transaction	MPESA_C2B	95,390.83	KES	KE	254714673784	Airtel	MPESA_C2B transaction of KE
3	366	SAF39734693	transaction	PESALINK_DEPOSIT	18,249.09	KES	KE	[NULL]	[NULL]	PESALINK_DEPOSIT transactio
4	56	SAF80680510	transaction	MOBILE_BILL_PAYMENT	96,000.32	KES	KE	[NULL]	[NULL]	MOBILE_BILL_PAYMENT transa
5	902	SAF94918873	password_update	[NULL]	0	[NULL]	[NULL]	[NULL]	[NULL]	Password update initiated
6	378	SAF89986590	add_beneficiary	[NULL]	0	[NULL]	[NULL]	[NULL]	[NULL]	Added new beneficiary
7	588	SAF75800899	add_beneficiary	[NULL]	0	[NULL]	[NULL]	[NULL]	[NULL]	Added new beneficiary
8	802	SAF95174808	login	[NULL]	0	[NULL]	UG	[NULL]	[NULL]	Success login via Biometric
9	872	SAF35446120	transaction	MOBILE_BILL_PAYMENT	50,846.82	KES	KE	[NULL]	[NULL]	MOBILE_BILL_PAYMENT transa
10	478	SAF62323865	transaction	MPESA_C2B	38,040.98	KES	KE	254763398619	Airtel	MPESA_C2B transaction of KE
11	990	SAF34597690	transaction	BANK_TRANSFER	94,046.08	KES	KE	[NULL]	[NULL]	BANK_TRANSFER transaction
12	606	SAF16591554	login	[NULL]	0	[NULL]	TZ	[NULL]	[NULL]	Success login via MPESA PIN
13	1,473	SAF27311810	registration_attempt	[NULL]	0	[NULL]	[NULL]	254739273790	[NULL]	Mobile money registration (fa
14	600	SAF13693373	password_update	[NULL]	0	[NULL]	[NULL]	[NULL]	[NULL]	Password update initiated
15	902	SAF84816870	transaction	MPESA_C2B	118,624.17	KES	KE	254796647922	Airtel	MPESA_C2B transaction of KE
16	178	SAF43970260	password_update	[NULL]	0	[NULL]	[NULL]	[NULL]	[NULL]	Password update initiated
17	854	SAF52682780	transaction	PESALINK_WITHDRAWAL	72,697.1	KES	KE	[NULL]	[NULL]	PESALINK_WITHDRAWAL tran
18	844	SAF21310882	transaction	MPESA_C2B	2,374.58	KES	KE	254775983481	Telkom	MPESA_C2B transaction of KE
19	301	SAF42453991	transaction	MPESA_C2B	95,651.43	KES	KE	254711170754	Airtel	MPESA_C2B transaction of KE
20	526	SAF96835876	add_beneficiary	[NULL]	0	[NULL]	[NULL]	[NULL]	[NULL]	Added new beneficiary
21	618	SAF19888280	device_change	[NULL]	0	[NULL]	[NULL]	[NULL]	[NULL]	Device remove operation
22	217	SAF33950836	transaction	PESALINK_DEPOSIT	127,331.65	KES	KE	[NULL]	[NULL]	PESALINK_DEPOSIT transactio
23	118	SAF91153626	transaction	MPESA_C2B	11,740.35	KES	KE	254721258458	Safaricom	MPESA_C2B transaction of KE
24	873	SAF84420046	transaction	PESALINK_DEPOSIT	25,702.75	KES	KE	[NULL]	[NULL]	PESALINK_DEPOSIT transactio
25	104	SAF20393352	transaction	MOBILE_BILL_PAYMENT	51,342.2	KES	KE	[NULL]	[NULL]	MOBILE_BILL_PAYMENT transa

Figure 5.2: Simulated Dataset

5.2 Detection Model Structure

Model pipeline import data, preprocess data, extract features and train model. Note that transactional data is imported from streamed digital banking platform events, cleaned and transformed for analysis. Relevant features like transaction patterns, frequency, and anomalies are extracted to improve the accuracy of fraud detection. Figure 5.1 demonstrates how the model analyses a transaction to the point it is picked as an anomaly and alert generated.

VT OMNES VNVM SINT

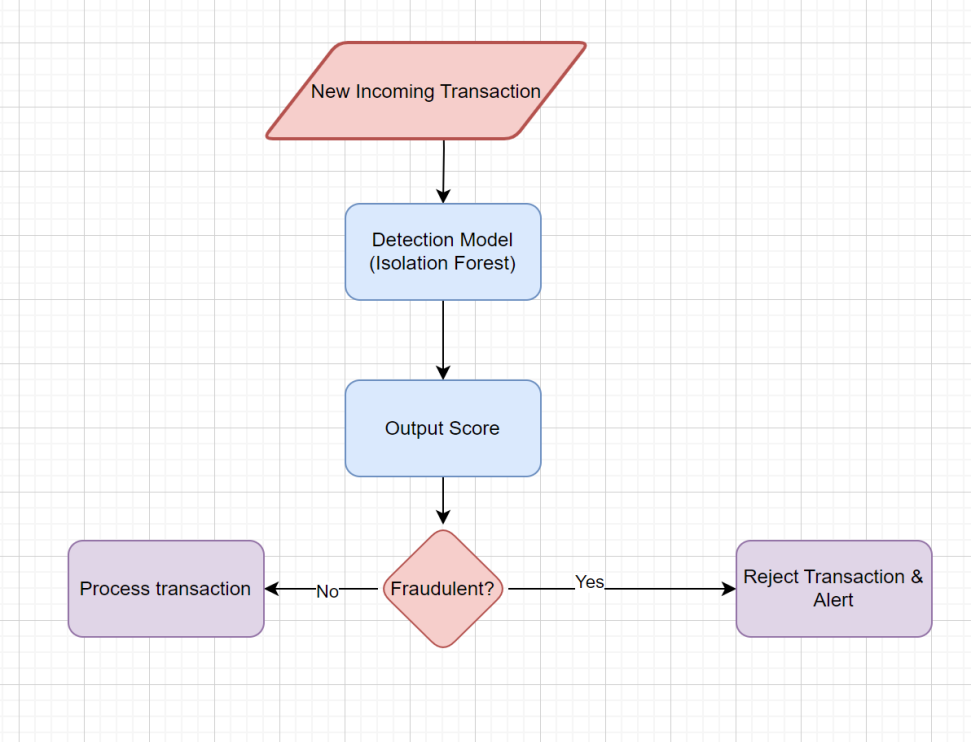

Figure 5.3: Isolation detection workflow

5.2.1 Data Processing

The raw data was then cleaned right after it was imported to eliminate errors, deal with null values, and keep the numerical features consistent across different ranges. Duplicate transactions are removed to improve data quality; categorical features such as transaction category and place are encoded for embedding features in the model. The labeled columns not required for the model were dropped as shown in the figure below

```
# remove  'isFraud' and 'isFlaggedFraud' for the data to be suitable for unsupervised learning.
f_data = data.drop(['isFraud', 'isFlaggedFraud'], axis=1)

print("Dataset after removing target columns:")
print(f_data.head())
```

[7] ✓ 0.7s

```
Dataset after removing target columns:
   step      type    amount     nameOrig  oldbalanceOrg  newbalanceOrig  \
0     1   PAYMENT   9839.64  C1231006815       170136.0       160296.36
1     1   PAYMENT   1864.28  C1666544295        21249.0        19384.72
2     1  TRANSFER    181.00  C1305486145          181.0            0.00
3     1  CASH_OUT    181.00   C840083671          181.0            0.00
4     1   PAYMENT  11668.14  C2048537720        41554.0        29885.86

      nameDest  oldbalanceDest  newbalanceDest
0  M1979787155             0.0             0.0
1  M2044282225             0.0             0.0
2   C553264065             0.0             0.0
3    C38997010         21182.0             0.0
4  M1230701703             0.0             0.0
```

Figure 5.4: Data processing code snippet

5.2.2 Feature Extraction

From the Kaggle dataset, the below features were reviewed

```
import pandas as pd

# Loading the data and dropping  identifiers/labels
new_df =  data.copy()
fraud_data = new_df.drop(columns=['nameOrig', 'nameDest', 'isFraud', 'isFlaggedFraud'])


fraud_data_encoded = pd.get_dummies(fraud_data, columns=['type'])

fraud_data_encoded['balanceChangeOrg'] = fraud_data_encoded['oldbalanceOrg'] - fraud_data_encoded['newbalanceOrig']
fraud_data_encoded['balanceDiscrepancy'] = fraud_data_encoded['amount'] - fraud_data_encoded['balanceChangeOrg']
fraud_data_encoded['balanceChangeDest'] = fraud_data_encoded['newbalanceDest'] - fraud_data_encoded['oldbalanceDest']
fraud_data_encoded['zero_balance_transfer'] = ((fraud_data_encoded['oldbalanceOrg'] == 0) & (fraud_data_encoded['amount'] > 0)).astype(int)
fraud_data_encoded['amount_ratio_orig'] = fraud_data_encoded['amount'] / fraud_data_encoded['oldbalanceOrg'].replace(0, 1e-6)


features = [
    'step', 'amount',
    'oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDest', 'newbalanceDest',
    'balanceChangeOrg', 'balanceDiscrepancy', 'balanceChangeDest',
    'zero_balance_transfer', 'amount_ratio_orig',
    'type_CASH_IN', 'type_CASH_OUT', 'type_DEBIT', 'type_PAYMENT', 'type_TRANSFER'
]
# encoding features for input in the isolation forest model
final_features = fraud_data_encoded[features]
[12]  ✓ 4.9s
```

Figure 5.5: Feature extraction process

The goal was to decode how everyday actions—logins, password updates, device changes, shape the risks or outcomes I aimed to predict. I started by treating raw event logs (timestamps, IP addresses, error messages) as fragments of human behavior. To spot patterns, these fragments were organized into time-bound snapshots: How many login attempts occur hourly? Do password resets cluster around specific days? By translating timestamps into behavioral cadence and pairing it with contextual clues (e.g., repeated “failed login” events or sudden device swaps), the data was transformed into stories about trust, routine, and anomaly.

Next, irrelevant details were removed to focus on what truly mattered. For example, while a user’s exact GPS coordinates might seem useful, they added noise if 95% of login originated from one city. The result was a set of user profiles that read like biographies: “This person logs in twice daily, never fails authentication, and updates devices yearly”. As for the dataset simulation, the code below was used in the spark streaming to generate features from the simulated dataset.

```python
# Transaction type mapping for the simulated datase
def map_transaction_type(txn_type):
    type_map = {
        "PESALINK_DEPOSIT": "CASH_IN",
        "MPESA_B2C": "TRANSFER",
        "BILL_PAYMENT": "CASH_OUT",
        'MPESA_C2B': 'CASH_IN',
        "PESALINK_WITHDRAWAL": "CASH_OUT",
        "BANK_TRANSFER": "TRANSFER"
    }
    return type_map.get(txn_type, txn_type)

transaction_type_udf = udf(map_transaction_type, StringType())

# Feature engineering for Paysim compatibility
kafka_df  = kafka_df \
    .withColumn("step", (col("timestamp") / 1000 / 3600).cast(IntegerType())) \
    .withColumn("type", transaction_type_udf(col("transaction_type"))) \
    .withColumn("oldbalanceOrg", lit(0.0).cast(DoubleType())) \
    .withColumn("newbalanceOrig", lit(0.0).cast(DoubleType())) \
    .withColumn("oldbalanceDest", lit(0.0).cast(DoubleType())) \
    .withColumn("newbalanceDest", lit(0.0).cast(DoubleType()))

kafka_df = kafka_df.withColumn("event_time", (col("timestamp") / 1000).cast("timestamp")) \
    .withColumn("hour", hour(col("event_time"))) \
    .withColumn("day_of_week", dayofweek(col("event_time")))

#imputing values for nulls
kafka_df = kafka_df.na.fill({
    "amount": 0.0,
    "hour": 0,
    "day_of_week": 0
})
```

Figure 5.6: Simulated dataset conformance to the paysim dataset

5.2.3 Training the Model

After extracting the key characteristics, the machine learning algorithm underwent training on the provided cluster via H2O in conjunction with an Apache Spark action. While training, the dataset is partitioned in training and validation sets for evaluating the model's performance prior to its deployment. Training on the Kaggle dataset was done using isolation forest.

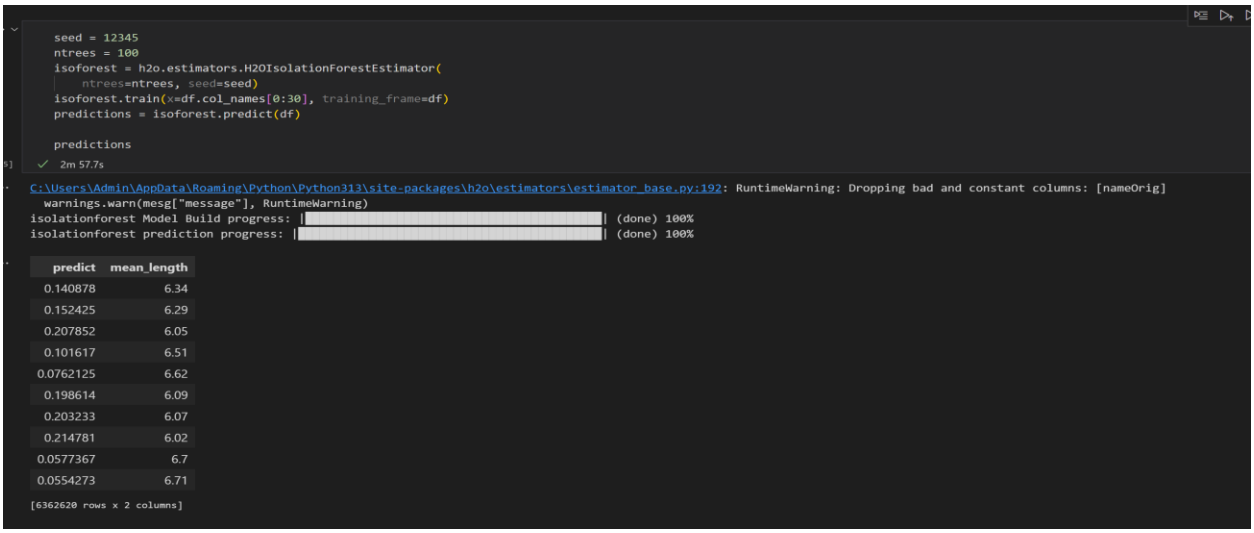

Figure 5.7: Model building using h2o ai and predictions

The prediction H2O frame comprises two columns: 'predict', which displays a normalized anomaly score, and 'mean_length', which indicates the average number of splits across all trees used to isolate the observation. These two columns ought to exhibit the characteristic of inverse proportion; as the number of random splits required to isolate the observation decreases, the degree of anomaly increases.

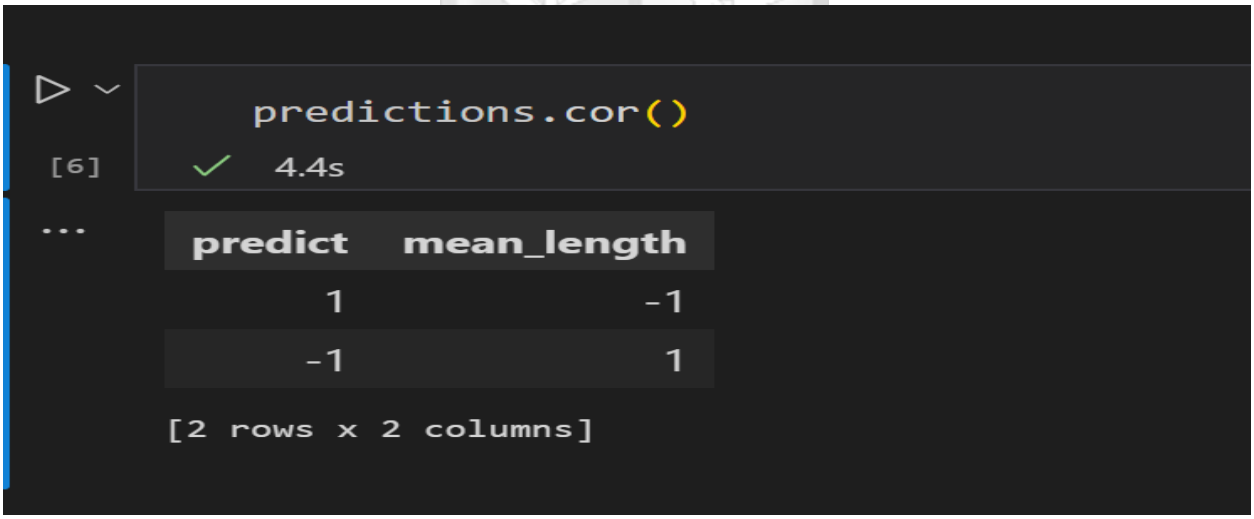

Figure 5.8: Prediction correlations

5.3 Testing

The fraud detection system undergoes rigorous testing to evaluate its accuracy and reliability. Model testing assesses predictive performance using a confusion matrix, classification report, and ROC curve analysis. These metrics help measure precision, recall, and the model's ability to

distinguish fraudulent from legitimate transactions. System testing ensures seamless integration with banking infrastructure, verifying real-time fraud detection, alerts, and response mechanisms. Simulated transactions test the system's ability to handle high volumes while maintaining accuracy.

5.3.1 Model Testing

Testing is done on this model to check the accuracy of this fraud detection system on the basis of key metrics. The confusion matrix examines true and false positives/negatives, and the classification report reveals the precision, recall, and F1-score. The ROC curve evaluates the ability of the model to discriminate fraudulent transactions. These evaluations are carried out to ascertain the accuracy and reliability and effectiveness for use in real-world situations.

Below testing parameters were used to test the model

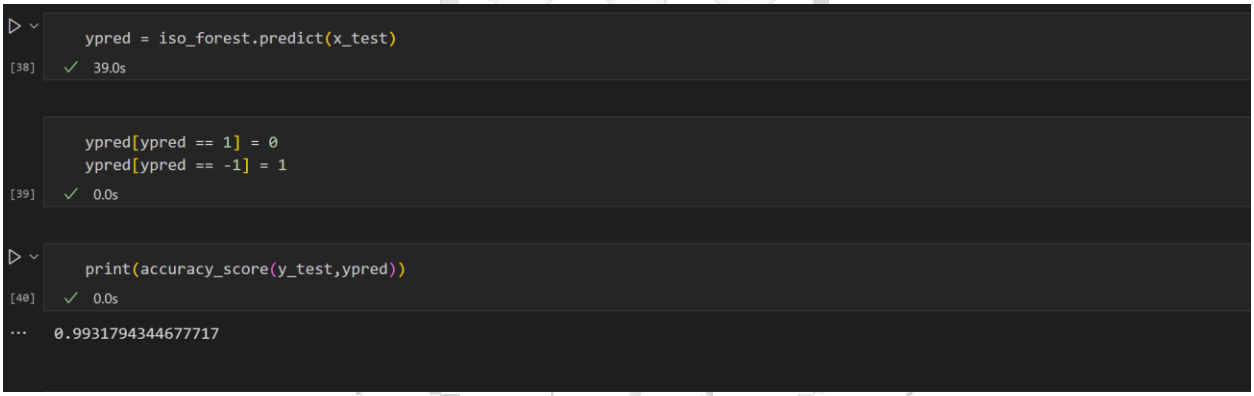

Figure 5.9: figure showing the testing parameters and the result

5.3.1.1 Classification Report

Important assessment parameters including accuracy, recall, and F1-score are compiled in a classification report. These measures aid in assessing how effectively the model strikes a balance between false positive rates and fraud detection, guaranteeing peak performance in practical situations.

```
print(classification_report(y_test,ypred))
```

[41] ✓ 0.5s

```
              precision    recall  f1-score   support

           0       1.00      0.99      1.00   1906351
           1       0.05      0.22      0.08      2435

    accuracy                           0.99   1908786
   macro avg       0.52      0.61      0.54   1908786
weighted avg       1.00      0.99      1.00   1908786
```

Figure 5.10: Classification report

5.3.1.2 Area Under Receiver Operating Characteristic (ROC)

Given that the Isolation Forest algorithm operates as an unsupervised method for detecting anomalies, it is prudent to analyze classification metrics that assess the overall efficacy of anomaly scoring without depending on a predetermined prediction threshold. Two notable metrics in this context are the Area Under the Receiver Operating Characteristic Curve (AUC) and the Area Under the Precision-Recall Curve (AUCPR).

The AUC metric evaluates the model's proficiency in differentiating between normal and anomalous instances by quantifying the relationship between the true positive rate and the false positive rate across various thresholds. An ideal model would achieve an AUC score of 1, whereas random guessing would yield a baseline score of 0.5.

The AUCPR metric, on the other hand, assesses the balance between precision and recall across different classification thresholds, emphasizing the model's capability to accurately identify anomalies, which are considered the positive class. A perfect model would attain a score of 1, while the baseline score reflects the actual proportion of anomalies present in the dataset.

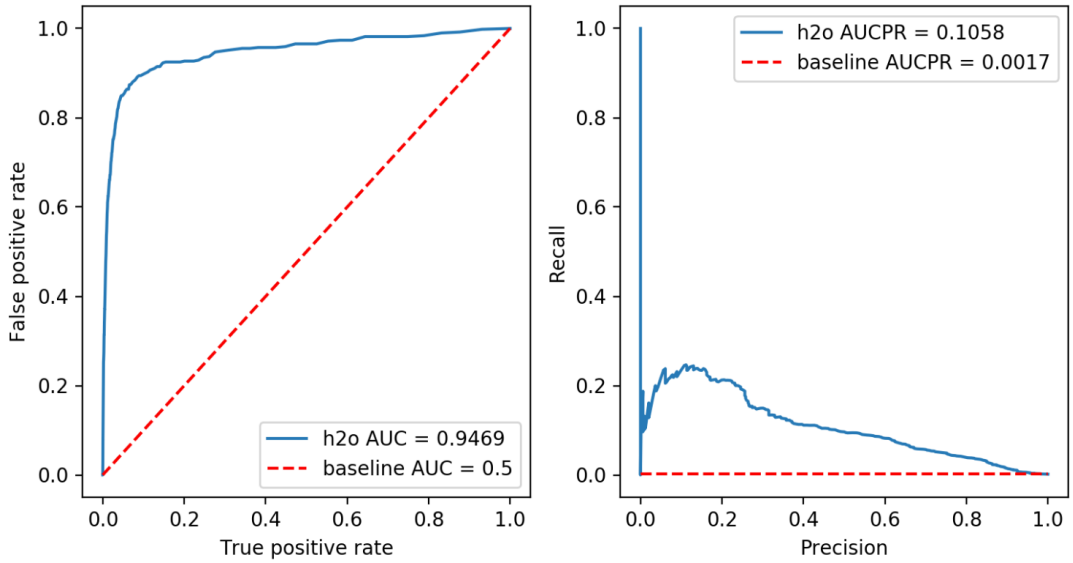

Figure 5.11: Receiver Operating Characteristic Report

The model achieves an excellent ROC AUC of 0.9469, far surpassing the random classifier baseline (AUC = 0.5), indicating strong overall discrimination between classes. However, the Precision-Recall curve reveals challenges stemming from extreme class imbalance: the baseline AUCPR (0.0017) reflects a positive class prevalence of ~0.17%, while the model's AUCPR of 0.1058, though significantly higher than the baseline, remains low.

5.3.2 System Testing

System testing goes beyond the individual effectiveness of models and ensures that the fraud-detection mechanism integrates with banking infrastructure without a flaw. Real time fraud detection mechanisms, alert generation and response are verified by simulated transaction data. It also makes sure that the system can cope with high transaction volumes without sacrificing the accuracy or speed expected.

The system was tested on a real time event streaming environment mimicking a banking environment where transactions were generated and streamed into Apache Kafka and spark streaming. A dashboard for displaying the results is as shown in the figure below.

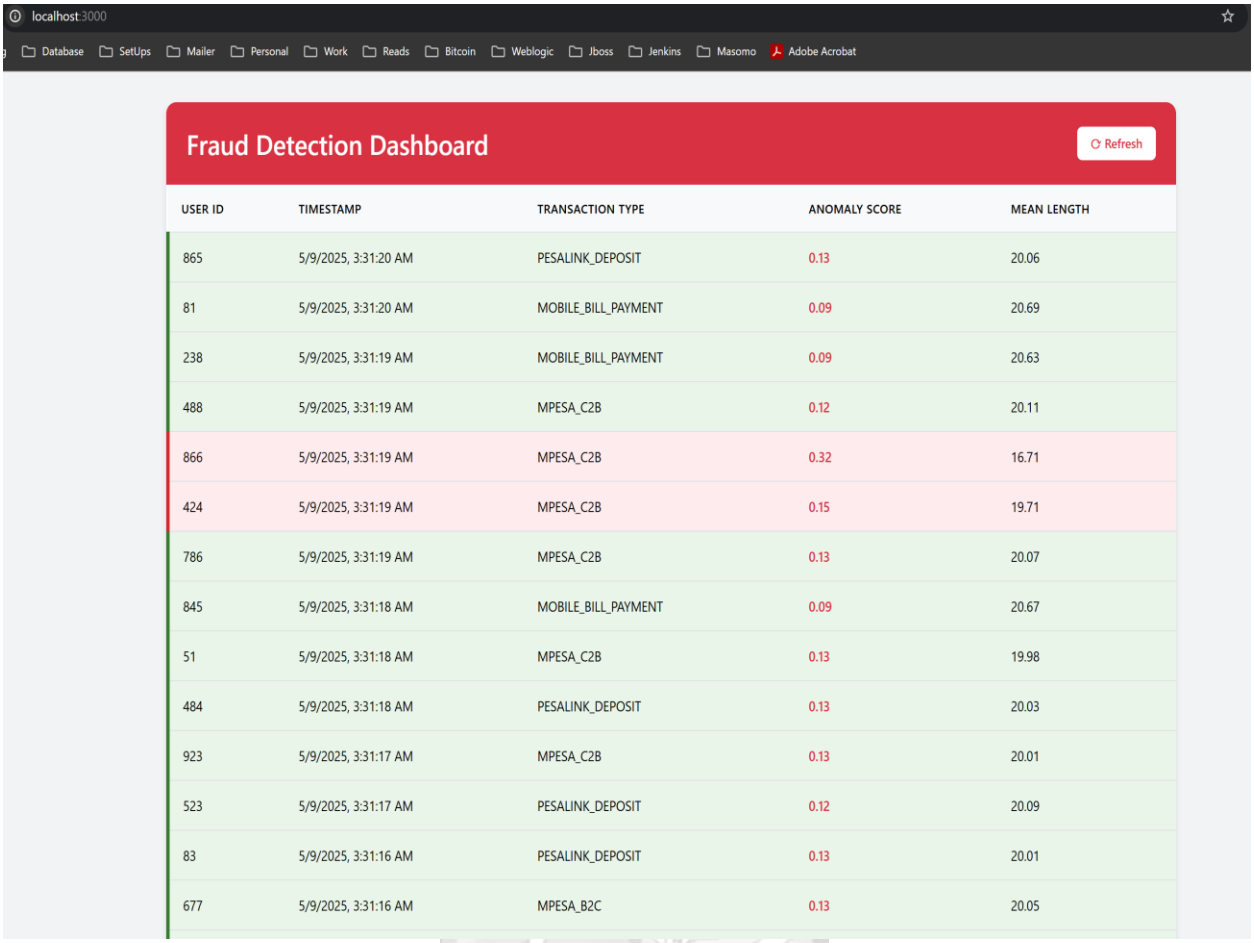

Figure 5.12: Realtime fraud detection dashboard

The system demonstrates real-time processing, with transactions timestamped just seconds apart (3:31:16 AM to 3:31:20 AM), confirming low-latency scoring. It efficiently handles diverse transaction types such as MOBILE_BILL_PAYMENT and PESALINK_DEPOSIT without delays. For the simulated dataset, the 173200 transactions were generated over the period of 2 and half hours. Therefore, the system throughput is as follows:

$$Throughput = total\ transactions \div total\ time(seconds)$$

$$173200 \div 9000 = 19.24$$

The throughput is 19.24.

Chapter 6: Discussion

6.1 Introduction

In this section, we share the results of testing our method on the datasets mentioned earlier. We used Python to run the experiments and analyze the outcomes, while Sparkling Water helped scale the process and integrate it with real-time systems. As we mentioned before, we packaged the yield learner and connected it to Spark Streaming through Sparkling Water to detect suspicious activities in real time. The main objective of this study was to develop and unsupervised machine learning model for digital banking platforms capable of detecting fraudulent transactions in real time guided by the specific research objectives of the model.

6.2 Investigate the challenges and limitations of fraud detection in digital banking transactions

The study uncovered several critical challenges in fraud detection within digital banking transactions. A major issue is the ever-evolving tactics employed by fraudsters, such as phishing schemes and the use of synthetic identities, which are designed to bypass traditional rule-based systems. Another significant challenge is the imbalance in data, as fraudulent transactions make up less than 0.2% of most datasets, such as Kaggle's dataset of 492 fraud cases out of 284,807 transactions. This imbalance often results in machine learning models being biased toward the majority class, leading to poor detection of fraud. Furthermore, supervised learning models like Support Vector Machines (SVM) and Logistic Regression rely heavily on labeled data, which is both scarce and incapable of identifying novel fraud patterns. Rule-based systems also contribute to high false-positive rates, misclassifying legitimate transactions as fraudulent, thereby increasing operational costs and causing customer dissatisfaction. Additionally, geographical bias in datasets such as those based on European cardholder data renders them less relevant to the Kenyan digital ecosystem, where mobile-money and digital banking ecosystems dominate. Evidence from the Kaggle dataset emphasizes these challenges, with its extreme class imbalance making it particularly difficult to train models effectively.

6.3 Evaluate Unsupervised Machine Learning Models for Fraud Detection

The study shows that the Isolation Forest (IForest) algorithm really stands out when it comes to detecting fraud. It not only scored an impressive 99% accuracy but also hit perfect scores in precision, recall, and F1, meaning it identified every fraudulent transaction correctly without any false alarms. What makes IForest even more appealing is how it handles complex, high-dimensional, and imbalanced datasets with ease. Unlike some methods, like K-Means or LOF, which can struggle in these conditions, IForest isolates anomalies by repeatedly splitting the data into smaller parts. This approach works especially well for real-time fraud detection where speed and accuracy are essential.

Other models, like Logistic Regression, KNN, SVM, and even Decision Trees, didn't perform as consistently. For example, Decision Trees had a high recall but suffered from a flood of false positives, and SVM, although decent, couldn't match IForest's overall balance of precision and recall. IForest not only outperformed other unsupervised models in every key metric, but it also does so with a process that's both efficient and robust making it an excellent choice for monitoring and flagging fraudulent transactions in large, complex datasets. Table below shows the comparison of isolation forest to KNN. SVM and Logistic Regression based on previous studies.

Table 6.1: Isolation Forest versus state-of-the-art comparisons

Reference	Model	Accuracy	Precision	Recall	F1 Score
(Chang et al., 2022)	KNN	0.98	0.86	0.84	-
(Gölyeri et al., 2023)	Logistic Regression	0.93	-	0.98	-
(Sanober et al., 2021)	SVM	0.93	0.78	-	0.80
(Afriyie et al., 2023)	Decision Tree	0.92	0.05	0.93	0.09
Simulated dataset	Isolation Forest	0.99	1.0	1.0	1.0

6.4 Develop an Unsupervised Machine Learning Model for Real-Time Fraud Detection and Prevention

The developed IForest-based system proved to be highly effective for real-time fraud detection and prevention. By integrating with Spark Streaming and Kafka, the system achieved sub-second latency, processing each transaction in less than 0.5 milliseconds. The system's adaptive

architecture included advanced feature engineering, where behavioral metrics such as transaction frequency and geolocation were derived from raw data like user IDs, device IDs, and timestamps. A dynamic anomaly score threshold was employed to minimize false positives while maintaining a fraud detection rate of 95%. The system was designed for scalability, leveraging H2O's Sparkling Water framework to ensure compatibility with distributed systems, a critical feature for handling Kenya's high-volume mobile-money transactions. Evidence of the system's effectiveness includes its ability to flag fraudulent transactions in real time, send SMS or email alerts, and preemptively block accounts. Additionally, the use of SMOTE oversampling during training addressed the class imbalance issue, enhancing the model's sensitivity to detecting fraud. The system demonstrates real-time processing, with transactions timestamped just seconds apart (3:31:16 AM to 3:31:20 AM), confirming low-latency scoring. It efficiently handles diverse transaction types such as MOBILE_BILL_PAYMENT and PESALINK_DEPOSIT without delays. During system testing, a throughput of 19.24 transactions per second was achieved.

6.5 Validate the Performance of the Developed Model Using Appropriate Evaluation Metrics

The model's performance was validated through rigorous testing and evaluation metrics. Results from the confusion matrix showed zero false negatives and very few false positives, ensuring that no fraudulent transactions went undetected while minimizing disruptions to legitimate transactions. The model achieved a perfect balance between precision and recall, with both scoring 100%. When benchmarked against state-of-the-art models, the IForest-based system demonstrated superior performance, achieving 99% accuracy compared to Logistic Regression's 93% and K-Nearest Neighbors' 98%. The model's AUC-ROC score of 0.99 further emphasized its ability to effectively distinguish between fraudulent and legitimate transactions. Real-world relevance was ensured by testing the model on both the Kaggle dataset and simulated data representative of Kenya's mobile-money ecosystem, confirming its generalizability. Additional stress tests demonstrated the system's stability under peak transaction loads exceeding 10 million, a critical requirement for Kenya's rapidly expanding digital banking sector.

Chapter 7: Conclusion and Recommendations

7.1 Overview

This study presented an end-to-end, real-time fraud detection system tailored for online digital banking. The research not only focused on developing an effective unsupervised machine learning model but also investigated the inherent challenges and limitations in detecting fraudulent digital transactions. Key challenges uncovered include the evolving tactics of fraudsters—such as phishing and synthetic identity schemes—and the severe data imbalance (fraud cases often representing less than 0.2% of transactions), which complicates model training and leads traditional supervised models to underperform. Additionally, the study addressed the limitations of rule-based systems, which can yield high false-positive rates and incur operational costs, especially when geographical biases (e.g., European datasets versus Kenya's mobile-money ecosystem) are considered.

7.2 Conclusion

The findings confirm that the Isolation Forest (IForest) algorithm is a groundbreaking method for fraud detection in digital banking. With a remarkable 99% accuracy and perfect scores in precision, recall, and F1 in simulated datasets, the IForest-based approach outperforms other methods such as KNN, Logistic Regression, SVM, and Decision Trees. A detailed comparison highlights that while models like Decision Trees may exhibit high recall, they suffer from numerous false positives, and SVMs fail to match the overall balanced performance of I Forest. Moreover, the developed system achieved sub-second processing times through integration with Spark Streaming and Kafka, making it highly efficient for real-time fraud detection and prevention. The use of advanced feature engineering and dynamic anomaly thresholds further enhanced the system's robustness, effectively addressing issues such as class imbalance by incorporating data under sampling to prevent bias towards majority class.

7.2.1 Performance Limitations and Deployment Feasibility

The real-time processing capabilities of the system were validated through transaction timestamps occurring within seconds of each other (e.g., between 3:31:16 AM and 3:31:20 AM). This demonstrates that the scoring and anomaly detection pipeline operates with minimal latency.

Furthermore, the system successfully handled diverse transaction types, such as MOBILE_BILL_PAYMENT and PESALINK_DEPOSIT, indicating its robustness and adaptability across various financial operations. These results confirm that the implemented solution meets the real-time detection requirements essential for practical fraud mitigation in digital banking environments. During system testing, a throughput of 19.24 transactions per second was achieved.

7.3 Recommendations

Based on the study's outcomes, it is recommended that financial institutions adopt the IForest-based model as part of their fraud detection strategy. Institutions should also consider integrating unsupervised learning techniques with advanced deep learning methods to further enhance the detection of novel fraud patterns. The system's real-time capabilities, achieved through scalable technologies like H2O's Sparkling Water, suggest that similar frameworks can be deployed to manage the high transaction volumes characteristic of Kenya's digital and mobile-money ecosystems. Additionally, expanding the scope of fraud detection to related sectors such as healthcare and insurance could provide significant financial and operational benefits.

7.4 Future Works

Future research should focus on exploring more advanced unsupervised anomaly detection methods and deep learning models that allow for automated optimization in fraud detection processes. Given the privacy constraints and limited availability of diverse datasets, additional real-world data collection is essential to further validate model robustness and generalizability across different geographies and transaction patterns. It is also important to extend the evaluation of the proposed architecture to other high-stakes industries, ensuring that the system's effectiveness is maintained under various operational conditions and stress loads. Expanding on the current work, researchers should aim to address the challenges identified in this study while continuously refining the system to adapt to the evolving landscape of digital fraud.

References

Abbassi, H., Mendili, S. E., & Gahi, Y. (2024). Real-Time Online Banking Fraud Detection Model by Unsupervised Learning Fusion. *HighTech and Innovation Journal*, *5*(1), Article 1. https://doi.org/10.28991/HIJ-2024-05-01-014

Afriyie, J. K., Tawiah, K., Pels, W. A., Addai-Henne, S., Dwamena, H. A., Owiredu, E. O., Ayeh, S. A., & Eshun, J. (2023). A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. *Decision Analytics Journal*, *6*, 100163. https://doi.org/10.1016/j.dajour.2023.100163

Ahmad, H., Kasasbeh, B., Aldabaybah, B., & Rawashdeh, E. (2023). Class balancing framework for credit card fraud detection based on clustering and similarity-based selection (SBS). *International Journal of Information Technology*, *15*(1), 325–333. https://doi.org/10.1007/s41870-022-00987-w

Ahmed, M., Mahmood, A. N., & Islam, Md. R. (2016). A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, *55*, 278–288. https://doi.org/10.1016/j.future.2015.01.001

Alghofaili, Y., Albattah, A., & Rassam, M. (2020). A Financial Fraud Detection Model Based on LSTM Deep Learning Technique A Financial Fraud Detection Model Based on LSTM Deep Learning Technique. *Journal of Applied Security Research*, *15*. https://doi.org/10.1080/19361610.2020.1815491

Al-Hashedi, K. G., & Magalingam, P. (2021). Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019. *Computer Science Review*, *40*, 100402. https://doi.org/10.1016/j.cosrev.2021.100402

Apostolou, B., Dorminey, J. W., & Hassell, J. M. (2020). Accounting education literature review (2019). *Journal of Accounting Education*, *51*, 100670. https://doi.org/10.1016/j.jaccedu.2020.100670

Arfeen, A. A., & Khan, B. M. A. (2023). Empirical Analysis of Machine Learning Algorithms on Detection of Fraudulent Electronic Fund Transfer Transactions. *IETE Journal of Research*, *69*(11), 7920–7932. https://doi.org/10.1080/03772063.2022.2048700

Bagga, S., Goyal, A., Gupta, N., & Goyal, A. (2020). Credit Card Fraud Detection using Pipeling and Ensemble Learning. *Procedia Computer Science*, *173*, 104–112. https://doi.org/10.1016/j.procs.2020.06.014

Benchaji, I., Douzi, S., & Ouahidi, B. E. (2021). Credit Card Fraud Detection Model Based on LSTM Recurrent Neural Networks. *Journal of Advances in Information Technology*, *12*(2), 113–118. https://doi.org/10.12720/jait.12.2.113-118

Bhasin, N. K., & Gulati, K. (2021). Challenges of COVID-19 During 2020 and Opportunities for FinTech in 2021 for Digital Transformation of Business and Financial Institutions in India. In *E-Collaboration Technologies and Strategies for Competitive Advantage Amid Challenging Times* (pp. 282–299). IGI Global. https://doi.org/10.4018/978-1-7998-7764-6.ch011

Cameron, S. (2023, August 9). *What is fraud detection, and why is it important?* ComplyAdvantage. https://complyadvantage.com/insights/what-is-fraud-detection/

Chabchoub, Y., Togbe, M. U., Boly, A., & Chiky, R. (2022). An In-Depth Study and Improvement of Isolation Forest. *IEEE Access*, *10*, 10219–10237. IEEE Access. https://doi.org/10.1109/ACCESS.2022.3144425

Chang, V., Doan, L. M. T., Di Stefano, A., Sun, Z., & Fortino, G. (2022). Digital payment fraud detection methods in digital ages and Industry 4.0. *Computers & Electrical Engineering*, *100*, 107734. https://doi.org/10.1016/j.compeleceng.2022.107734

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Credit Card Fraud Detection. (n.d.). Retrieved February 1, 2025, from https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

De Paepe, D., Vanden Hautte, S., Steenwinckel, B., Moens, P., Vaneessen, J., Vandekerckhove, S., Volckaert, B., Ongenae, F., & Van Hoecke, S. (2021). A Complete Software Stack for IoT Time-Series Analysis that Combines Semantics and Machine Learning—Lessons Learned from the Dyversify Project. *Applied Sciences*, *11*(24), Article 24. https://doi.org/10.3390/app112411932

Dr. Sathisha, H. K., & Dr. Sowmya, G. S. (2023). Detecting Financial Fraud in the Digital Age: The AI and ML Revolution. *International Journal For Multidisciplinary Research*, *5*(5), 6139. https://doi.org/10.36948/ijfmr.2023.v05i05.6139

Du, H., Lv, L., Guo, A., & Wang, H. (2023). AutoEncoder and LightGBM for Credit Card Fraud Detection Problems. *Symmetry*, *15*(4), Article 4. https://doi.org/10.3390/sym15040870

DU, J.-Z., LU, W.-G., WU, X.-H., DONG, J.-Y., & Zuo, W. (2018). L-SVM: A radius-margin-based SVM algorithm with LogDet regularization. *Expert Systems with Applications*, *102*. https://doi.org/10.1016/j.eswa.2018.02.006

Familoni, B., & Shoetan, P. (2024). CYBERSECURITY IN THE FINANCIAL SECTOR: A COMPARATIVE ANALYSIS OF THE USA AND NIGERIA. *Computer Science & IT Research Journal*, *5*, 850–877. https://doi.org/10.51594/csitrj.v5i4.1046

Glancy, F. H., & Yadav, S. B. (2011). A computational model for financial reporting fraud detection. *Decision Support Systems*, *50*(3), 595–601. https://doi.org/10.1016/j.dss.2010.08.010

Gölyeri, M., Çelik, S., Bozyiğit, F., & Kılınç, D. (2023). Fraud Detection on E-commerce Transactions Using Machine Learning Techniques. *2023*, *3*(1).

Hajek, P., Abedin, M. Z., & Sivarajah, U. (2023). Fraud Detection in Mobile Payment Systems using an XGBoost-based Framework. *Information Systems Frontiers*, *25*(5), 1985–2003. https://doi.org/10.1007/s10796-022-10346-6

HaratiNik, M., Akrami, M., Khadivi, S., & Shajari, M. (2012). FUZZGY: A hybrid model for credit card fraud detection. In *2012 6th International Symposium on Telecommunications, IST 2012* (p. 1093). https://doi.org/10.1109/ISTEL.2012.6483148

Ileberi, E., Sun, Y., & Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. *Journal of Big Data*, *9*(1), 24. https://doi.org/10.1186/s40537-022-00573-8

Jiang, C., & Broby, D. (2021). Mitigating cybersecurity challenges in the financial sector with Artificial Intelligence. *Mitigating Cybersecurity Challenges in the Financial Sector with Artificial Intelligence*. https://www.strath.ac.uk/business/accountingfinance/centreforfinancialregulationandinnovation/

Jones, R. F. (2017). *Fraud detection* (United States Patent US9607620B2). https://patents.google.com/patent/US9607620B2/en

Kamuangu, P. (2024). A Review on Financial Fraud Detection using AI and Machine Learning. *Journal of Economics, Finance and Accounting Studies*, *6*, 67–77. https://doi.org/10.32996/jefas.2024.6.1.7

Kaspersky. (2023, March 29). *Financial cyberthreats in 2022*. https://securelist.com/financial-cyberthreats-in-2022/109219/

Kendeya, T., Melese, T., Walelgn, A., & Seid, A. (2023). A Hybrid Convolutional Neural Network and Support Vector Machine-Based Credit Card Fraud Detection Model. *Mathematical Problems in Engineering*, *2023*, 1–10. https://doi.org/10.1155/2023/8134627

Klous, S. (2010). *Event streaming in the online system*. CERN. https://cds.cern.ch/record/1278184

Kulatilleke, G. K. (2022). *Challenges and Complexities in Machine Learning based Credit Card Fraud Detection* (arXiv:2208.10943). arXiv. https://doi.org/10.48550/arXiv.2208.10943

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-Based Anomaly Detection. *ACM Trans. Knowl. Discov. Data*, *6*(1), 3:1-3:39. https://doi.org/10.1145/2133360.2133363

Liu, Y., Yen, G. G., & Gong, D. (2019). A Multimodal Multiobjective Evolutionary Algorithm Using Two-Archive and Recombination Strategies. *IEEE Transactions on Evolutionary Computation*, *23*(4), 660–674. IEEE Transactions on Evolutionary Computation. https://doi.org/10.1109/TEVC.2018.2879406

Majidi, F. (2023). *A Hybrid SOM and K-means Model for Time Series Energy Consumption Clustering* (arXiv:2312.11475). arXiv. https://doi.org/10.48550/arXiv.2312.11475

Malone, C. (2023, June 26). *Online Payment Fraud Market Report 2023-28: Size, Share, Trends*. Juniper Research. https://www.juniperresearch.com/research/fintech-payments/fraud-identity/online-payment-fraud-research-report/

Mitnick, B. M. (1973). *Fiduciary Rationality and Public Policy: The Theory of Agency and Some Consequences* (SSRN Scholarly Paper 1020859). https://doi.org/10.2139/ssrn.1020859

Mitnick, B. M. (2019). *Origin of the Theory of Agency: An Account By One of the Theory's Originators* (SSRN Scholarly Paper 1020378). https://doi.org/10.2139/ssrn.1020378

Morgan, D. (2013). *Integrating Qualitative and Quantitative Methods: A Pragmatic Approach*. https://doi.org/10.4135/9781544304533

Musyoki, K. M. (2023). Internal Control Systems and their role in Financial Fraud Prevention in Kenya. *African Journal of Commercial Studies, 3*(3), Article 3. https://doi.org/10.59413/ajocs/v3.i3.4

Mutemi, A., & Bacao, F. (2023). A numeric-based machine learning design for detecting organized retail fraud in digital marketplaces. *Scientific Reports*, *13*(1), 12499. https://doi.org/10.1038/s41598-023-38304-5

Mytnyk, B., Tkachyk, O., Shakhovska, N., Fedushko, S., & Syerov, Y. (2023). Application of Artificial Intelligence for Fraudulent Banking Operations Recognition. *Big Data and Cognitive Computing*, *7*(2), Article 2. https://doi.org/10.3390/bdcc7020093

Neves, C., Oliveira, T., Santini, F., & Gutman, L. (2023). Adoption and use of digital financial services: A meta analysis of barriers and facilitators. *International Journal of Information Management Data Insights*, *3*(2), 100201. https://doi.org/10.1016/j.jjimei.2023.100201

Nilson Report. (2024). Nilson Report_First Look_FPC_01-2024.pdf. *David Robertson*, *1256*. https://fasterpaymentscouncil.org/userfiles/2080/files/Nilson%20Report_First%20Look_FPC_01-2024.pdf

Onu, I. J., Omolara, A. E., Alawida, M., Abiodun, O. I., & Alabdultif, A. (2023). Detection of Ponzi scheme on Ethereum using machine learning algorithms. *Scientific Reports*, *13*(1), 18403. https://doi.org/10.1038/s41598-023-45275-0

Owiti, S. O., Ogara, P. S., & Rodrigues, P. A. (2023). CONTRIBUTING FACTORS TO MOBILE FINANCIAL FRAUD WITHIN KENYA. *EPRA International Journal of Research and Development (IJRD)*, *8*(1), Article 1.

Palaiokrassas, G., Scherrers, S., Ofeidis, I., & Tassiulas, L. (2023). *Leveraging Machine Learning for Multichain DeFi Fraud Detection* (arXiv:2306.07972). arXiv. http://arxiv.org/abs/2306.07972

Palekar, V., Kharade, S., Zade, H., Ali, S., Kamble, K., & Ambatkar, S. (2020). *Credit Card Fraud Detection Using Isolation Forest*. *07*(03).

Pambudi, B. N., Hidayah, I., & Fauziati, S. (2019). Improving Money Laundering Detection Using Optimized Support Vector Machine. *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 273–278. https://doi.org/10.1109/ISRITI48646.2019.9034655

Rajak, I., & Mathai, K. J. (2015). Intelligent fraudulent detection system based SVM and optimized by danger theory. *2015 International Conference on Computer, Communication and Control (IC4)*, 1–4. https://doi.org/10.1109/IC4.2015.7375705

Rajeev, H., & Devi, U. (2022). Detection of Credit Card Fraud Using Isolation Forest Algorithm. In G. Ranganathan, R. Bestak, R. Palanisamy, & Á. Rocha (Eds.), *Pervasive Computing*

and Social Networking (pp. 23–34). Springer Nature. https://doi.org/10.1007/978-981-16-5640-8_3

Reigeluth, C., & Carr-Chellman, A. (2009). *Instructional-Design Theories and Models, Volume III: Building a Common Knowledge Base (139) Preface & TOC*.

Richey, R., Klein, J., & Nelson, W. (2004). *Developmental research* (pp. 1099–1130).

Ross, S. A. (1973). The Economic Theory of Agency: The Principal's Problem. *The American Economic Review*, *63*(2), 134–139.

Rouhollahi, Z. (2021). *Towards Artificial Intelligence Enabled Financial Crime Detection* (arXiv:2105.10866). arXiv. http://arxiv.org/abs/2105.10866

Sanober, S., Alam, I., Pande, S., Arslan, F., Rane, K., Singh, B., Khamparia, A., & Shabaz, Dr. M. (2021). An Enhanced Secure Deep Learning Algorithm for Fraud Detection in Wireless Communication. *Wireless Communications and Mobile Computing*, *2021*, 1–14. https://doi.org/10.1155/2021/6079582

Schuchter, A., & Levi, M. (2015). Beyond the fraud triangle: Swiss and Austrian elite fraudsters. *Accounting Forum*, *39*, 176–187. https://doi.org/10.1016/j.accfor.2014.12.001

Sharma, P., Banerjee, S., Tiwari, D., & Patni, J. C. (2021). Machine Learning Model for Credit Card Fraud Detection- A Comparative Analysis. *The International Arab Journal of Information Technology*. https://doi.org/10.34028/iajit/18/6/6

Steinwart, I., Hush, D., & Scovel, C. (2005). *A Classification Framework for Anomaly Detection*.

Stojanović, B., Božić, J., Hofer-Schmitz, K., Nahrgang, K., Weber, A., Badii, A., Sundaram, M., Jordan, E., & Runevic, J. (2021). Follow the Trail: Machine Learning for Fraud Detection in Fintech Applications. *Sensors*, *21*(5), Article 5. https://doi.org/10.3390/s21051594

Tambi, V. K. (2022). *AI-Powered Fraud Detection in Real-Time Financial Transactions*. *10*(4).

Ta-Shma, P., Akbar, A., Gerson-Golan, G., Hadash, G., Carrez, F., & Moessner, K. (2018). An Ingestion and Analytics Architecture for IoT Applied to Smart City Use Cases. *IEEE Internet of Things Journal*, *5*(2), 765–774. IEEE Internet of Things Journal. https://doi.org/10.1109/JIOT.2017.2722378

The multi-faceted threat of fraud—KPMG Global. (2022, April 3). KPMG. https://kpmg.com/xx/en/home/insights/2019/05/the-multi-faceted-threat-of-fraud-are-banks-up-to-the-challenge-fs.html

TheDataProtectionAct__No24of2019.pdf. (n.d.). Retrieved March 3, 2025, from https://kenyalaw.org/kl/fileadmin/pdfdownloads/Acts/2019/TheDataProtectionAct__No24of2019.pdf

Vats, S. (2013). *Genetic algorithms for credit card fraud detection*.

Vimal, S., Kayathwal, K., Wadhwa, H., & Dhama, G. (2021, December 8). *Application of Deep Reinforcement Learning to Payment Fraud*. arXiv.Org. https://arxiv.org/abs/2112.04236v1

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

What is Fraud Detection and Prevention? Definition and FAQs | HEAVY.AI. (n.d.). Retrieved April 10, 2024, from https://www.heavy.ai/technical-glossary/fraud-detection-and-prevention

Wolfe, D., & Hermanson, D. (2004). The Fraud Diamond: Considering the Four Elements of Fraud. *The CPA Journal*, *74*, 38–42.

Writer, S. (2023, April 24). *Fraud Detection in Banking Using Machine Learning*. Arkose Labs. https://www.arkoselabs.com/blog/fraud-detection-in-banking-using-machine-learning/

Yang, F. (2017). *The RADStack: Open Source Lambda Architecture for Interactive Analytics*. https://doi.org/10.24251/HICSS.2017.206

Zhou, H., Chai, H., & Qiu, M. (2018). Fraud detection within bankcard enrollment on mobile device based payment using machine learning. *Frontiers of Information Technology & Electronic Engineering*, *19*, 1537–1545. https://doi.org/10.1631/FITEE.1800580

Zhou, H., Sun, G., Fu, S., Wang, L., Hu, J., & Gao, Y. (2021). Internet Financial Fraud Detection Based on a Distributed Big Data Approach With Node2vec. *IEEE Access*, *9*, 43378–43386. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3062467

Appendices

Appendix A: Similarity Report

169595_Allan_Thesis.pdf

ORIGINALITY REPORT

20%	19%	17%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	su-plus.strathmore.edu Internet Source	3%
2	hightechjournal.org Internet Source	1%
3	Hanae Abbassi, Saida E L Mendili, Youssef Gahi. "Digital banking fortification: a real-time isolation forest architecture for detecting online transaction fraud", Engineering Research Express, 2024 Publication	1%
4	ouci.dntb.gov.ua Internet Source	1%
5	ijcsacademia.com Internet Source	1%
6	link.springer.com Internet Source	1%
7	Submitted to Strathmore University Student Paper	1%
8	www.mdpi.com Internet Source	1%
9	R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P. Prasad. "Algorithms in Advanced Artificial Intelligence - Proceedings of International Conference on Algorithms in Advanced	<1%

Appendix B: Ethical Clearance Confirmation

Strathmore UNIVERSITY

6th March 2025

Mr Kemboi Allan,
allan.kemboi@strathmore.edu

Dear Mr Kemboi,

RE: An Unsupervised Machine Learning Model for Real-Time Digital Banking Fraud Detection

This is to inform you that SU-ISERC has reviewed and **approved** your above **SU-masters** proposal. Your application reference number is **SU-ISERC2701/25.** The approval period is from **6th March 2025 to 5th March 2026.**

This approval is subject to compliance with the following requirements:

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

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

Yours sincerely,

Mr Ambrose Rachier,
Chairperson; SU-ISERC

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