University of Essex

School of Mathematics, Statistics
and Actuarial Science

MA981 DISSERTATION

ANOMALY DETECTION FOR MOBILE MONEY FRAUD TRANSACTIONS

KAGO RONALD THIPE

Supervisor: **Dr TAO GAO**

September 18, 2024
Colchester

Abstract

Innovative solutions are required to minimise financial loss and improve the speed of catching fraud transactions. This dissertation investigates the application of supervised machine learning models for detecting fraud in mobile money transactions. With the rapid global growth of mobile financial services, fraud detection has become an essential aspect of safeguarding these platforms. The study evaluates supervised models, including logistic regression, random forests, and neural networks, using a synthetic dataset that simulates mobile money transactions. The performance of each model is assessed based on accuracy, true positive rate, false positive rate, and computational efficiency. The findings show that XGBoost and logistic regression offers a strong balance between accuracy and computational speed, while neural networks provide higher accuracy but at a significant computational cost. This study concludes by highlighting the importance of deploying supervised models in real-time fraud detection systems and recommending further research to optimize model performance for practical applications.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

As of 2024, $1.75 billion registered mobile money accounts are processing $1.4 trillion a year, which translates to $2.7 million per minute [3]. Although significant progress has been made financially including the global population, approximately 20% of the global population, or 1.6 billion people, remain unbanked [5]. It is essential to note that trust remains a significant barrier to the adoption of digital financial services. According to a survey mentioned in the Economic Times, a notable proportion of users, particularly in underdeveloped communities, prefer using cash over digital transactions due to concerns about security and a lack of trust in the Internet and digital platforms [4]. For many, the perception that someone else could gain access to their bank accounts is a significant deterrent to using digital payment systems. This lack of trust poses a challenge to telecommunications companies and other mobile money providers, who must not only create accessible mobile money platforms but also ensure that these platforms are perceived as safe and secure by their users. According to [14] $33.5 billion in 2023 was reported to be lost in card fraud around the world, and overall financial losses due to card fraud are expected to keep going up to, reaching $43.47 billion in 2028. In North America $466 million were reported as fraudulent while in Europe €1.53 billion in card transactions were reported as fraudulent in 2023. So if fully functioning market systems still experience losses due to fraudulent actions, developing markets such as Africa and Asia need to ensure that they have systems in place that try to reduce Financial Fraud Transactions and in turn protect both the business and the customers.

This dissertation explores the application of machine learning models for detecting

anomalous transactions that could indicate fraud in mobile money operations. The focus on anomaly detection is driven by the need to quickly and accurately identify potentially fraudulent activities, thereby safeguarding the integrity of mobile money platforms. The research aims to answer the following key questions:

1. What are the most effective supervised machine learning methods for anomaly detection in mobile money transactions?
2. How do the selected models perform in terms of accuracy, true positive rate, false positive rate, and computational efficiency?
3. Which supervised models offer the best balance between detection performance and real-time applicability for mobile money fraud detection systems?
4. What are the key factors or considerations in determining the overall success and reliability of a fraud detection system beyond standard performance metrics?

The structure of this dissertation is organised as follows: Chapter 2 provides an overview of mobile money operations and the existing challenges in anomaly detection. Chapter 3 focuses on the methodologies used in anomaly detection, with a particular focus on fraud detection in financial transactions. Chapter 4 presents the data analysis, including pre-processing and pattern recognition, essential for effective anomaly detection. Chapter 5 discusses implementing various machine learning techniques and evaluates their performance. Finally, Chapter 6 concludes the research by summarizing the findings and suggesting areas for future research.

CHAPTER 2

Literature Review

2.1 Overview of Mobile Money Operations

Suri [2] defined Mobile Money as a platform that enables mobile phone owners to deposit, transfer and withdraw funds without owning a bank account. Customers are able to use Mobile Money agents in their localities instead of using Bank Branches or Bank ATM's for digital financial transactions. The state of the Mobile Money Industry by GSMA [3] indicates that 70% of Mobile Money accounts are found in Sub-Saharan Africa as seen in figure 2.1 below. As a positive, Mobile Money has also contributed positively to countries where Mobile Money operates, contributing over $600 billion[3].

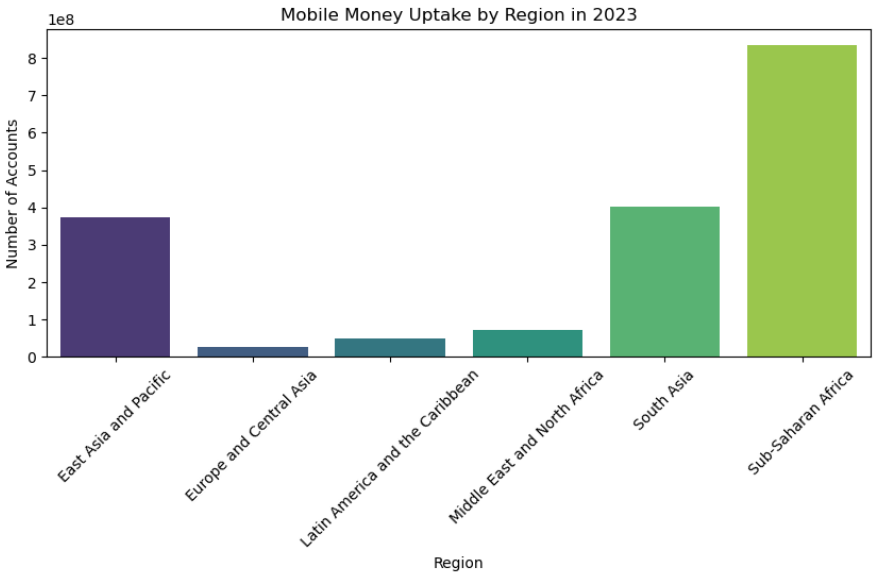


Figure 2.1: Graph showing Sub-Saharan Africa's leadership in mobile money adoption, highlighting the region's prominence in global mobile financial services.
Source: GSMA, State of Mobile Money 2023.

Mobile money has also done great in reducing the gap of the global population who are not financially included. The World Bank [5] considers financial inclusion as a key component in reducing global poverty. They further indicate that ownership of a financial account is the first step towards a broader financial inclusion since people can now store money, and send and receive money. The number of adults without a financial account has decreased from 2.5 billion in 2011 to 1.4 billion in 2021, and the majority of great strides have been due to the existence of Mobile Money. Figure 2.2 shows that since 2011, Mobile Money accounts have been exponentially increasing.

Besides just the normal financial transactions that can be done with an account, Mobile Money is now transforming to being a fully digital platform [3]. Some of those services include taking a loan, paying for insurance, International remittances, saving money, etc. However [1] argues that cross-border transactions present a great threat to law enforcement in the fight against money laundering. The lack of hindrances to creating a Mobile Money account can be also seen as a disadvantage as there is a slightly less strict rule regarding who is sending or receiving money, and Money Laundering can be a form of financial fraud existing in this industry. There are also many other financial means of fraud which we will discuss in 2.3 in detail.

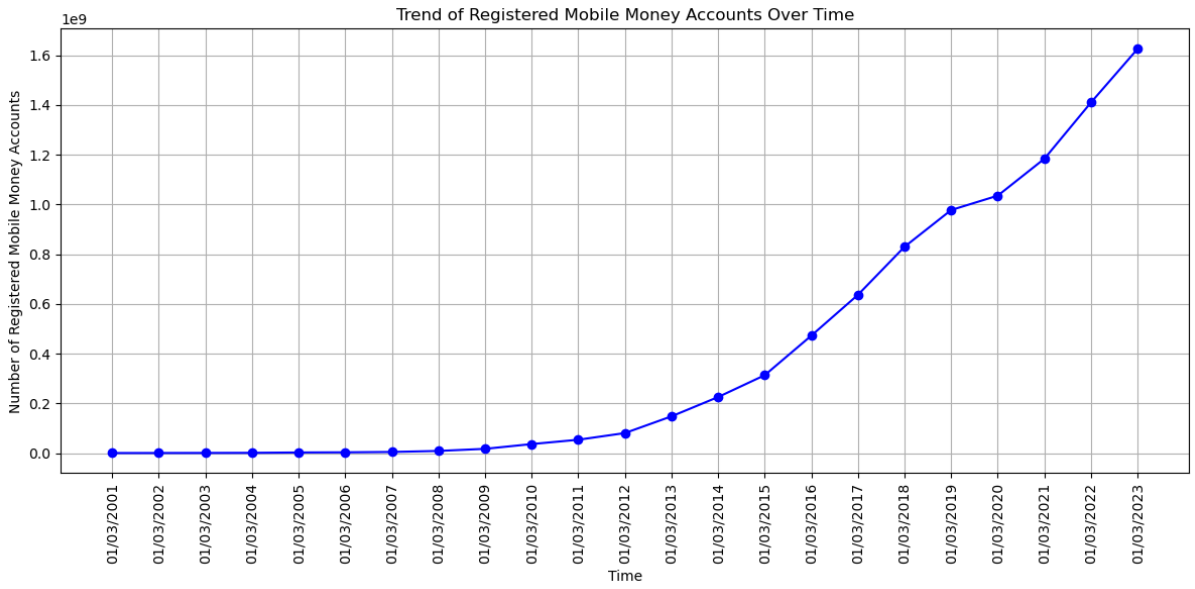


Figure 2.2: Graph illustrating the growth trajectory of mobile money services over time, emphasizing the rapid expansion of digital financial platforms.
Source: GSMA, State of Mobile Money 2023.

2.2 Anomaly Detection

[6] defines Anomaly Detection as the problem of finding patterns in data that do not conform to expected behaviour. There are various areas where Anomaly Detection can be applied, and some of these areas are Credit Card Fraud transactions, Insurance, health care, cyber security, military surveillance for enemy activities and other areas. This is because in these areas we often expect subjects to follow a rather normal behaviour, and if at any instance the behaviour changes, then that might be an anomaly and it is worth an investigation. For example, if you are used to taking a certain route to work/ school using whatever mode of transport allows you to be on time every day for your first lesson/ work meeting but let us say you try to take your route and there is a protest or roadwork on your route thus delaying your work and making you take a different a route, and for the first time you are late, wouldn't you want to have a system that detects that therefore allowing you to prepare well, this is what anomaly detection is for. In this paper we will use Outlier and Anomaly interchangeably as they both mean [6] patterns in data that do not conform to a well-defined notion of normal behaviour, but we are however interested in that anomaly that can have dire effects and almost usually

caused by humans and hence Fraud. In Statistics, outliers can be defined as Extreme Values or values easier to spot in a Histogram [7].

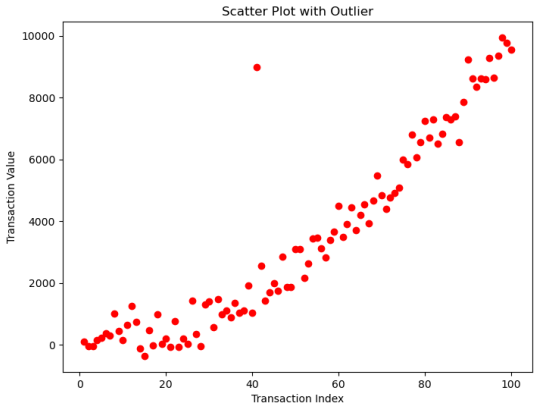


Figure 2.3: Example of an outlier in a scatter plot, demonstrating the visual detection of anomalies in financial transaction data

[8] said an outlier is a point that falls more than 1.5 times the interquartile range(IQR), while [9] said one way to think of outliers is that they are inconsistent with the remainder of the observations in the dataset.

However, when trying to deal with outliers, there exist some challenges. [6] included :

- difficulty in defining a normal region that encompasses every possible normal behaviour.
- After some time, anomalous observations can appear normal.
- anomalies are different for different domains.
- availability of labelled data for training models

Because Mobile Money is still a developing industry, such challenges might still exist. For example, if a person is running a fraudulent account with malicious activity, it might become difficult to flag transactions from such an account as Fraudulent, the same thing also applies to doing International transactions.

Anomaly detection has been applied in different areas, and some of them are :

- Intrusion detection: detection of malicious activities in a computer-related system. The key challenge here is the high volumes of data, making it difficult to individually analyse false alarms.

- Fraud detection: We are working on Fraud detection for Mobile Money Transactions, but fraud can exist in banks, credit cards, insurance, telecommunications, the stock market etc. Fraud exists when users or non-users(identity theft) consume the resources provided by the organization in an unauthorized way. The challenge with this is that it requires detection to occur as soon as the fraudulent transaction takes place to avoid economic losses.

- Medical Anomaly Detection: This is regarding patient data, and anomalies may happen due to abnormal patient conditions, instrumentation or recording errors. Or sometimes detecting a disease outbreak. The main challenge in this field is the cost of classifying an anomaly as normal is too high, and in some instances it is life-threatening.

So far there have been multiple techniques used in trying to detect anomalies, most of them using Statistical or Machine Learning techniques, and these techniques can be divided into the below sections:

- Classification Based: the dataset is labelled with data instances that can be classified as normal or anomalous,and it has two phases. The training phase learns a classifier using the available labelled training data, then the test phase uses the learned model to classify a test instance as either normal or anomalous. The biggest assumption is that the normal and anomalous classes can be learned in a given feature space. These methods we can use for our Mobile Money Transactions given that we have a dataset that already has There are different, and we will be looking at some of the methods in our Data Analysis.

- Nearest-Neighbourhood Based: Here normal data instances occur in dense neighbourhoods, while anomalies that occur are far from their closest neighbours.

- Clustering Based: Normal data instances belong to a cluster in the data, while anomalies do not belong to any cluster. Although clustering techniques are not optimized to find anomalies, an assumption to be made is that normal instances

are closer to a cluster centroid, while anomalies are far from their closest cluster centroid.

- Statistical Methods: Normal data instances occur in high-probability regions of a stochastic model, while anomalies occur in the low-probability regions of the stochastic model.
- Information Theory: Anomalies in data induce irregularities in the information content of the data set.
- Spectral Techniques: Data can be embedded into a lower dimensional subspace in which normal instances and anomalies appear significantly different.

2.3 Fraud in Financial Transactions

Financial Fraud can take many different forms [10], which could be false financial disclosures, financial scams and financial mis-selling. Reurink [10] further goes on to define Financial Fraud as acts and statements through which financial market participants in the market deliberately or recklessly provide them with false, incomplete or manipulative information related to financial goods, services and investment opportunities in a way that violates any kind of legal stipulation, be it regulatory rule, statutory law, civil law, or criminal law.

Financial Fraud acts are different depending on the market they are performed on, the actors and the regulatory operations in place in an environment. For example in Botswana [11], multiple Botswana people lost millions to a Ponzi scheme which claimed they could just download money from the internet every day not knowing that the catch was them referring other people, and paying to upgrade to the next level. It might seem ridiculous that people can fall for such scams, but we have to understand that in such markets, victims have limited knowledge regarding internet scams, and some are even desperate for easy money given the high rate of unemployment in such countries [12]. Orange Money, a Mobile Money Operator was one of the financial platforms the scammers used to receive and give payments to victims, and because of the clueless nature of the victims, the responsibility is now on Mobile Money Operators to ensure such a fraudulent activity does not occur or at least it does not occur at such a high

level.

Fraud has got different impact on the community, and some are not only financial but rather can have considerable social and psychological effects on individuals, businesses, and society. The impacts of fraud can also be broken down into different types [13]. Human Impact whereby victims, family members of victims, and the community are left with a traumatic experience. It should be noted that most individuals who suffer the most are from disadvantaged or vulnerable environments. Fraud can have a reputational impact for example on the Financial Organisation, or the Government especially if not handled right. The EU [15] constantly updates its list of high-risk third-country jurisdictions presenting deficiencies in their anti-money laundering systems. Being on such a list might deprive a country of access to financial services. Other impacts could be Business and Industrial related.

CHAPTER 3

Formulations/Methodologies

Due to the lack of publicly available datasets on Mobile Money services, the dataset used is a synthetic dataset generated using a PaySim Simulator [16]. PaySim [16] is a financial simulator that simulates mobile money transactions based on an original dataset. The dataset used in this study is a 30-day simulation, which corresponds to 744 time steps (measured in hours).Such a dataset was created for Researchers to build to research on topics like Fraud, Mobile Money uptake using a dataset that mimics real-world data.The sample dataset used in this study can be found at the following link: Kaggle PaySim Dataset.We will be running our Machine Learning Models on this dataset to ensure that we can somehow improve the prediction of fraudulent transactions. The structure of the dataset is as such :

Field	Description
step	Represents a unit of time where 1 step is 1 hour.
type	Type of transaction e.g., 'PAYMENT', 'TRANSFER', 'CASH OUT'.
amount	Amount of the transaction.
nameOrig	Customer who initiated the transaction.
oldbalanceOrg	Initial balance before the transaction.
newbalanceOrig	New balance after the transaction.
nameDest	Recipient of the transaction.
oldbalanceDest	Initial balance of the recipient before the transaction.
newbalanceDest	New balance of the recipient after the transaction.
isFraud	Whether the transaction is fraudulent (1) or not (0).
isFlaggedFraud	Whether the transaction was flagged as fraudulent by the system (1) or not (0).

Table 3.1: Descriptions of features used in the Mobile Money Transactions dataset, including transaction types, balances, and fraud indicators

Data Preprocessing, analysing and modelling were all carried out using Python 3.12 programming language. The Integrated Development Environment was Jupyter Notebooks. The computational hardware used was a Lenovo Thinkpad equipped with an Intel Core i5-7200uCPU @ 2.50GHz 2.71 GHzG7 processor,8 GB of RAM, and a 512 GB SSD for storage. It is important to note that computational speed may vary depending on the hardware configuration used.

3.1 Anomaly Detection Methods to be Used

This study exclusively utilizes supervised machine learning methods due to their proven effectiveness in identifying fraud based on labelled data. The models employed include Logistic Regression, Random Forest, and Neural Networks, among others, which are evaluated on their ability to detect fraudulent mobile money transactions. Unsupervised methods were initially considered but excluded due to their computational cost and the synthetic nature of the dataset, which made supervised methods more practical for this

study.

[19] explains that supervised learning involves a model that estimates an output based on one or more inputs. For our case, our dataset has a variable *isFraud* which identifies transactions or instances that were anomalous and hence fraud. Supervised Machine Learning methods act as an improvement to existing rule-based methods as they are effective in terms of time and costs, and they are less prone to false-positive results [1]. For supervised methods to work, we need to have already identified old instances of transactions that have been identified as fraud so that our models can learn from the historical data to accurately predict the response/behaviour of future transactions.

Some of the supervised methods we will be using based on their effectiveness in previous studies of fraud detection [18] are:

- k-Nearest Neighbour (k-NN)
- Random Forests
- XGBoost
- Logistic Regression
- Naive - bayes
- Neural Networks

The implementation of these methods follows a similar pattern, where we have to preprocess the data, and this could involve handling missing values, normalizing features and converting categorical predictor variables to numerical ones. We then have to split the dataset into train and test sets, implement the classification model, train the model on the training data then evaluate performance on the test data. Another big challenge we had to overcome was dealing with the imbalance of the classification variable, and we had to apply SMOTING for that.

Synthetic Minority Over-sampling Technique (SMOTE) [17] is a method used to address class imbalance in datasets. In fraud detection, the number of fraudulent transactions is often much smaller than the number of legitimate transactions, leading to difficulty in accurately training machine learning models. SMOTE works by generating

synthetic samples of the minority class (fraudulent transactions) based on existing data. It creates new instances by interpolating between the nearest neighbours of minority class samples, thus increasing the representation of the fraud class without merely duplicating existing data. This technique helps the model learn better and improve its performance in detecting fraud.

k-NN

Research indicates that k-Nearest Neighbors (k-NN) or sometimes called Instance-Based Learning is a well-suited method for fraud detection due to its simplicity and effectiveness in identifying outliers or anomalies in datasets. In fraud detection, especially in financial systems, k-NN is used to classify transactions as either fraudulent or legitimate by comparing them to known cases. The method relies on the assumption that fraudulent transactions will appear as outliers when compared to the majority of legitimate transactions in the dataset. k-NN is particularly advantageous for fraud detection because it is an instance-based learning method, which means it does not assume an underlying distribution for the data. Instead, it directly compares new transactions to historical data, making it highly flexible and adaptable to different types of fraud. Additionally, k-NN's ability to work with minimal assumptions makes it powerful in scenarios where fraudulent behaviour is unpredictable or varies significantly from known patterns. k-NN has been deployed in various fraud detection beyond financial services such as Credit Card Fraud detection cite, Insurance Fraud Detection cite, and Telecommunications Fraud [18].

k-NN operates by storing all the training examples and, for any new example, classifies it based on the majority class among its k most similar (nearest) examples in the training data [21]. It assumes that a meaningful distance metric can be defined between examples and that cases close to each other in the feature space are likely to belong to the same class. However, since it requires storing all training examples and searching through them for classification, it can be computationally expensive, especially with large datasets like the one we worked with. Since also all features are treated equally, it can perform poorly if irrelevant features are present and this is why we had to ensure that we perform an informed feature selection.

The classification decision in k-NN can be expressed mathematically as:

$$\hat{y} = \arg \max_{c \in \text{Classes}} \sum_{i=1}^{k} Q(y_i, c) \tag{3.1}$$

Where $Q(y_i, c) = 1$ if $y_i = c$, and $Q(y_i, c) = 0$ otherwise. The goal is to predict the class c that maximizes the number of neighbors among the k closest examples that belong to class c. In the context of fraud detection, c represents the "isFraud" class, and the decision function would select the class that most neighbours in the training set belong to.

Random forest

Random Forests is an ensemble learning technique that operates by constructing multiple decision trees during training and outputting the mode of the classes for classification. This method is particularly robust in handling high-dimensional data and provides good accuracy for fraud detection task [18]. Random Forest is highly suitable for fraud detection because it aggregates the decisions of multiple decision trees, leading to more accurate and stable predictions. It can handle both categorical and continuous variables, making it versatile for different types of fraud detection [1]

XGBoost

[20] focused their research solely on XGBoost and why it is effective for fraud detection. They indicated that XGBoost is a highly optimized and scalable implementation of gradient boosting. It performs well in cases of extreme class imbalance, which is as common for fraud detection datasets as it is for our case. XGBoost builds an ensemble of decision trees in a stepwise manner to minimize the overall prediction error by introducing models that correct the errors of previous models. It assumes that by sequentially adding models, each new model improves the overall prediction accuracy. It can however be computationally intensive, especially with large datasets, due to the need for multiple rounds of model training.

Logistic Regression

[1] used Logistic Regression as a baseline model to compare with other techniques for Mobile Money Fraud detection techniques. Although it might not always outperform more complex models, it remains a baseline model for fraud detection as it has been

used also in detecting credit card [18], insurance fraud [18], and medical billing fraud [18] with reliable results. Logistic regression is advantageous as it is highly scalable, can handle large datasets and is easy to interpret. This makes it simple to deploy and maintain. However, if we do not effectively preprocess the data, Logistic Regression can overfit the model, and it might also struggle with understanding non-linear relationships in the data. Mathematically, we can define Logistic Regression as [19] :

$$Pr(Y = 1|X = x) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}} \tag{3.2}$$

Where:

- Y is the binary outcome (fraud or no fraud),
- X represents the independent features,
- β_0 and β_1 are the model coefficients (Applied AI Letters - 20...).

Neural networks

Neural Networks are possibly one of the most powerful tools for fraud detection and this is due to their ability to model complex, non-linear relationships in the data [1]. They are highly effective as they can capture complicated patterns and correlations in large datasets, which might be missed by simpler models. A Neural Network is a Machine Learning model inspired by the human brain consisting of layers of interconnected nodes (neurons). These networks can learn complex functions by adjusting the weights of connections based on the input data, allowing them to predict outcomes such as whether a transaction is fraudulent or not.

Bayesian Classifier

For Bayesian, we will be using a Naive Bayes Classifier which is good for fraud detection tasks due to its ability to handle large datasets efficiently and with simplicity [18]. Naive Bayes operates under the principle of calculating the probability of a transaction being fraudulent based on the conditional probability of certain features, given the class (fraud or non-fraud). The most critical assumption however is **Conditional Independence**, as it assumes the presence of a particular feature in a class is independent of any other feature, given the class label. This simplifies computation but it might not hold

true in real-world data. The classification decision in Naive Bayes can be expressed mathematically as:

$$\hat{y} = \arg \max_{c \in \text{Classes}} P(c) \prod_{i=1}^{n} P(x_i|c) \tag{3.3}$$

Where:

- $P(c)$ is the prior probability of class c.
- $P(x_i|c)$ is the likelihood of feature x_i given class c.
- The product is taken over all features x_i.

In the context of fraud detection, this formula helps calculate the posterior probability that a transaction belongs to the "isFraud" class, given the observed features of the transaction.

The other model which we struggled to run due to its high computational cost was Support Vector Machines. Previous research did indicate that SVM are good for fraud detection questions.

3.2 Evaluating performance of The Anomaly Detection Methods

In the context of evaluating anomaly detection methods, selecting the right performance metrics is crucial to understanding the effectiveness of the models. For this dissertation, the performance metrics chosen are Accuracy, True Positive Rate (Recall), and False Positive Rate (FPR). These metrics are particularly relevant for assessing the performance of fraud detection models, given the specific challenges posed by the data.

Accuracy is a commonly used performance metric that measures the overall correctness of a model by calculating the proportion of correctly classified instances out of the total instances [19]. In the domain of fraud detection, accuracy is important because it provides a broad overview of how well the model is performing across both fraudulent and legitimate transactions. However, it's important to note that accuracy alone can be misleading, especially in the case of imbalanced datasets, which are common in fraud detection. For example, if the dataset contains 99% legitimate transactions and 1%

fraudulent transactions, a model that classifies all transactions as legitimate would have a high accuracy rate but would fail entirely at detecting fraud. Thus, while accuracy is a useful starting point, it must be considered alongside other metrics that provide more granular insights [20]

The True Positive Rate, or Recall, is a critical metric in fraud detection. It measures the proportion of actual fraud cases that are correctly identified by the model. Specifically, Recall is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.4}$$

where:

- TP (True Positives) are the cases where the model correctly identifies fraud.
- FN (False Negatives) are the cases where the model fails to identify fraud.

In fraud detection, Recall is particularly important because missing a fraudulent transaction (a false negative) can have significant financial implications. High Recall indicates that the model is effective at capturing fraudulent activities, even if it means increasing the number of false positives. This trade-off is often acceptable in fraud detection scenarios where the cost of missed fraud is much higher than the cost of investigating false alarm [20]

Another metric we will be using is False Positive Rate which measures the proportion of legitimate transactions that are incorrectly classified as fraudulent by the model. It is calculated as:

$$\text{FPR} = \frac{FP}{FP + TN} \tag{3.5}$$

where:

- FP (False Positives) are the legitimate transactions incorrectly flagged as fraud.
- TN (True Negatives) are the legitimate transactions correctly identified as non-fraudulent.

Minimizing the FPR is crucial in maintaining a positive customer experience, as legitimate transactions flagged as fraudulent can lead to customer dissatisfaction and

potential loss of business. Therefore, while high Recall is desirable, it is also important to keep the FPR as low as possible to avoid unnecessary disruptions to legitimate users [20]

We chose the above as performance metrics due to the need to balance the trade-offs between detecting as many fraudulent transactions as possible and minimizing the impact on legitimate users. In fraud detection, especially in mobile payment systems, this balance is critical. High Accuracy ensures that the model is generally reliable, while high Recall ensures that most fraudulent transactions are caught. At the same time, a low FPR helps maintain customer trust and satisfaction by reducing the number of false alarms.

Another metric we have chosen to investigate is the computational speed of running a specific model. This is because we want to choose a model that can deliver quick responses with minimal disruptions, which is effective if the model is to be deployed fro real time fraud detection systems.

CHAPTER 4

Data analysis

4.1 Data Pre-processing and Cleaning

We will be performing a variety of data preprossessing depending on the time of the analysis we will be doing. Fortunately, our dataset does not have any missing values, which means we do not have to handle any missing values cases. Although this is a synthetic dataset, the lack of missing values does mimic real world dataset unless there is a problem with the system, but well functioning should always have a record of every customer's transaction.

One-hot Encoding will also be applied to the Type column to convert the categorical column to a numerical column. All columns being numerical makes our machine learning models more accurate and easier to run. We will also be removing the column isFlaggedFraud as every transaction that isFlaggedFraud is also indicated asFraud. For our supervised learning models, we will also be removing the columns oldbalanceOrg, newbalanceOrig, oldbalanceDest, newbalanceDest as Transactions which are detected as Fraud have been cancelled, therefore in this situation you will realise that oldBalanceOrg is equal to Amount, which is not really accurate.

4.2 Patterns and Trends Observed in the Dataset

By performing different Exploratory Data Analysis methods we can observe the distribution of transaction types, amounts and correlations between feauters. Exploring our dataset also allows us to identify the hiddern patterns in our dataset which is necessary for outlier detection, as we can get to know what to expect and we also get to summarise insights that might possibly be held in more than a million rows.

4.2.1 Distribution of Transaction Types

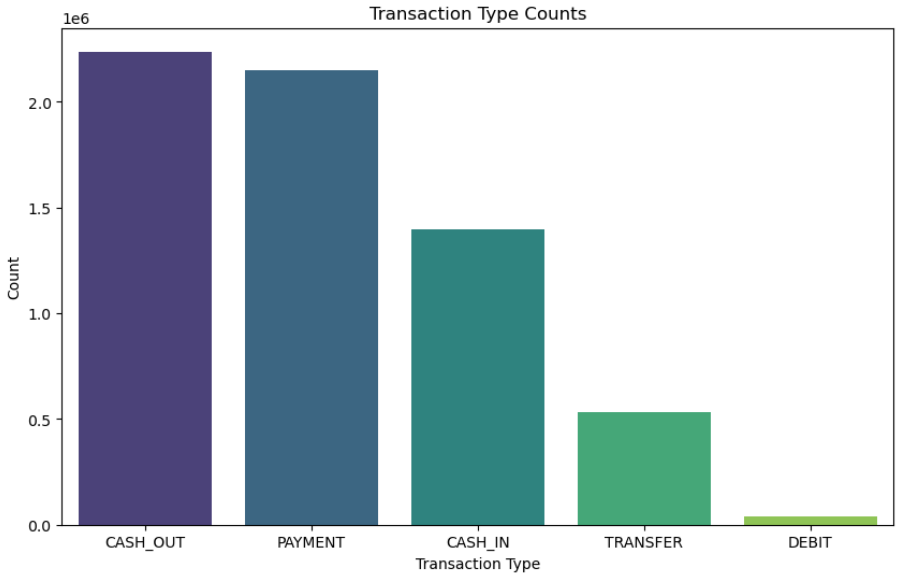


Figure 4.1: Bar chart showing the distribution of different transaction types (e.g., cash-outs, payments) in the mobile money dataset

From the above chart, it is prominent that "Cash-Outs" Transaction Types followed by "Payment" are the leading transaction types. This is an expected behaviour as Mobile Money was mainly created to assist the unbanked to transact digitally. This basically means that a banked individual will send cash to an unbanked individual, and the unbanked will probably cash out the money and transact with cash, or perform payments with money directly on the mobile money platform. It is also not surprising to see Debit transaction types being low, as debit cards are seen as an advanced concept, especially in rural areas, and there are few ATMs in those areas, which does translate to "Debit" transaction types. Knowledge of such information is important for anomaly

detection as an unusual activity such as a rise in debit payments can be an alert for investigations. This is also important as Mobile Money Operators need to know their customers. One way fraudsters carry out their fraud is them cashing out money at multiple ATMs and within a short period of time especially since ATMs have a limit on how much to withdraw.

4.2.2 Distribution of Transaction Amounts

As a way of also understanding how customers behaviour, how money moves within businesses, merchants and customers, it is important that we identify accounts that might hold a huge amount of money, the average money we expect to be in people's accounts, and an alarming amount that might be found in an individual account.

Transaction Amount Range	**Count**
0-100,000	3,525,298
100,001-200,000	1,163,752
200,001-500,000	1,333,286
500,001-1,000,000	209,658
1,000,001-92,445,516	130,626

Table 4.1: Distribution of transaction amounts across different value ranges in the Mobile Money dataset.

Although not explicit from the chart above, the median amount in an individual's account is 74,871. There are however anomalous accounts that have more than 10 million in their accounts. Knowledge of such information could help an organisation create measures for controlling fraud especially if the system cannot detect yet. We are interested in learning the customer's behaviour, whether they transact a lot, or transact with few money then detect if their accounts have been attacked by fraudsters. A majority of customers have got less than 100,000, and this can inform the normal behaviour of customers using the Mobile Money platform.

4.2.3 Correlation Matrix

Another interesting plot is the correlation matrix. It is meant to reveal the relationship between different numerical features in the dataset. Such a plot is also important for fraud detection as we can be aware how different features influence each other.

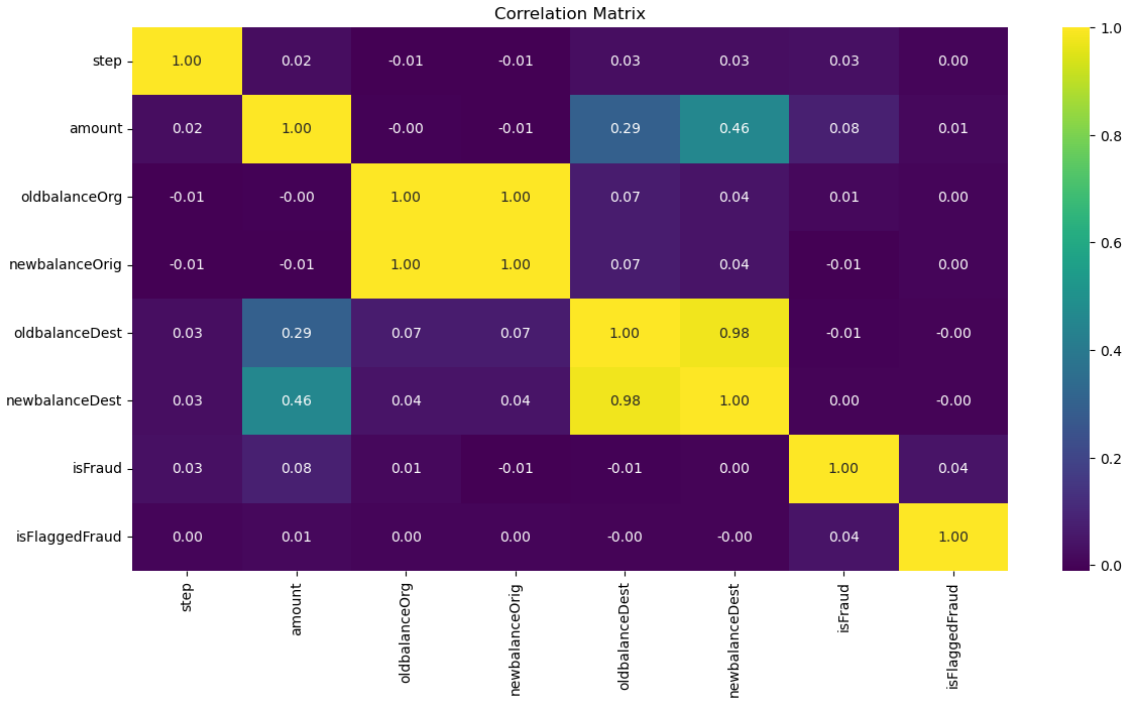


Figure 4.2: Correlation matrix showing relationships between different numerical features in the mobile money dataset, identifying potential influences on transaction behavior

From the plot, we can see that there is Strong negative correlation between 'oldbalanceOrg' and 'newbalanceOrg'. This is an expected behaviour as Money has to come from one account and flows to another individual's account. An issue were money cannot be traced might mean that our systems as an organisation are not capable of identifying fraud and our data collection systems are not effective.

4.2.4 Distribution of Fraudulent Transactions

isFraud	Count
0	6,354,407
1	8,213

Table 4.2: Summary of fraudulent and non-fraudulent transactions in the dataset, showing the imbalance between the two classes.

As expected, Fraudulent should only occur on rare occasions. This is also what makes them so difficult to learn and prepare for, but the effects of one fraudulent transaction can cost the economy millions. In the table above, we can see that there is an uneven distribution of transactions recorded as fraud to those which are legitimate, this might make it difficult for our algorithms to learn fraud better as it will only have a few examples of what fraud is. Within our Machine Learning, we will apply different concepts such as Smoting, to address the uneven distribution of fraud class.

One of the things, we need to take care of when assessing fraud is that there are certain periods of time when transactions go up, this could be due to an increase in seasonal spending, for example, the Christmas holidays. And for this reason, we have to ensure to our algorithms are able to identify when such goes on and do not flag such transactions as fraudulent. We can see in this plot that at around the time step of about 300, there is a spike in transaction amounts. Is this due to fraud, or is it a normal behaviour?

The legitimate transactions follow the trend provided by all transactions, this might be partly due to the fact that the dataset is mostly made by legitimate transactions. But the we can see that the fraudulent do not really have a known pattern, that is because fraud is unpredictable in regards to time. It can happen anytime. Therefore

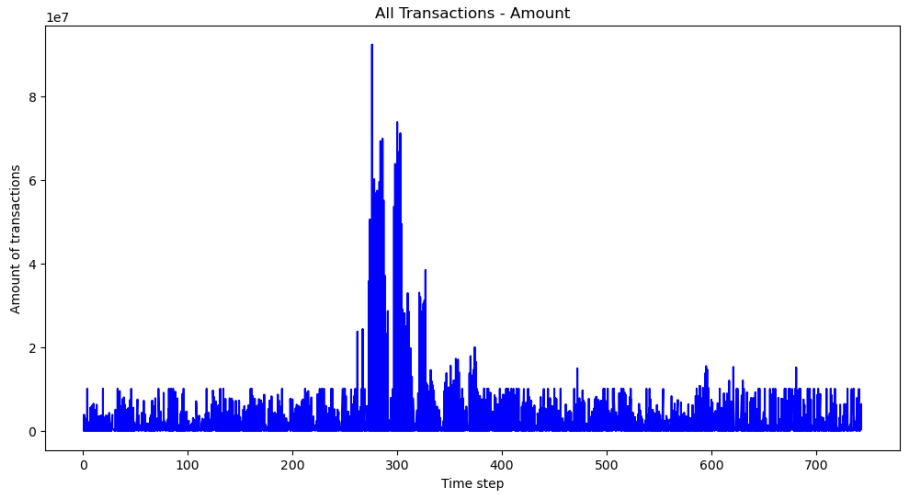
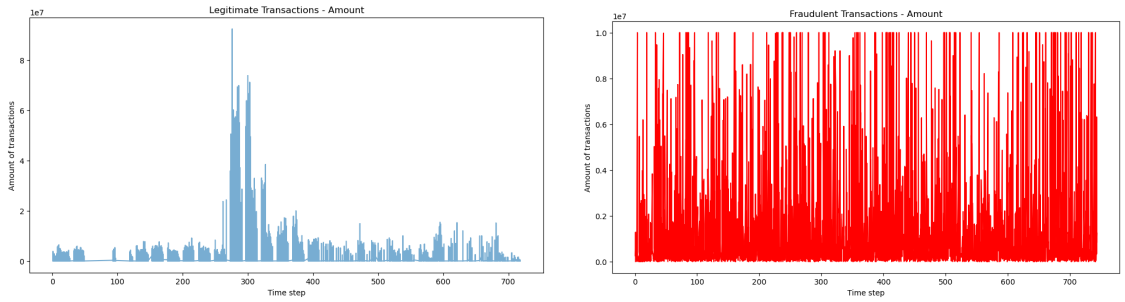


Figure 4.3: Time series plot showing the total transaction amounts over time, indicating periods of high and low financial activity



(a) Legitimate Transactions

(b) Fraudulent Transactions

Figure 4.4: Time series comparison of legitimate versus fraudulent transactions, illustrating the unpredictability of fraud occurrence in mobile money systems

feeding a model that predicts fraud, we need to look at other factors like the volume of transactions, frequency of transactions, recipient of transactions etc to ensure we predict accurately.

CHAPTER 5

Implementation of Anomaly Detection Techniques

5.1 Supervised Learning Methods

Having a dataset that already has a column with previously trained examples of fraud, we can use classification systems to train.

Table 5.1: Performance comparison of various supervised machine learning models based on true positive rate (TPR), false positive rate (FPR), accuracy, and computational speed

Model	TP	FP	Accuracy	Computational Speed
Logistic Regression	95.86%	5.08%	94.92%	61.79s
Naive Bayes Classifier	100%	43.44%	56.62%	4.19s
Neural Network	98.78%	0.62%	99.38%	3256.32s
K-NN	82.55%	0.44%	99.54%	118.14s
Random Forest	96.19%	0.07%	99.92%	4682.67s
XGBoost	98.99%	0.25%	99.75%	99.08s

In evaluating the performance of various anomaly detection methods for fraud detection, we not only consider traditional metrics such as True Positive Rate (TP), False Positive Rate (FP), and Accuracy but also incorporate Computational Speed into our analysis [18]. This addition is critical as it ensures that the selected model can deliver quick responses with minimal disruptions, which is essential for real-time fraud

detection systems.

Logistic Regression shows a strong overall performance with a high True Positive Rate of 95.86% and a manageable False Positive Rate of 5.08%. The accuracy of 94.92% indicates that this model is reliable in detecting fraudulent activities while keeping the number of false alarms relatively low. Furthermore, the computational speed of 61.79 seconds suggests that Logistic Regression is fast enough for real-time fraud detection scenarios, making it a balanced choice between performance and speed [23].
Naive Bayes achieves a perfect True Positive Rate of 100%, meaning it successfully identifies all fraudulent transactions in the dataset. However, this comes at the cost of a very high False Positive Rate of 43.44%, leading to a low overall accuracy of 56.62%. While it is computationally very efficient, with a speed of just 4.19 seconds, the trade-off between speed and accuracy may not justify its use in environments where minimizing false positives is critical. This is consistent with findings that Naive Bayes classifiers can be efficient but may not be the most effective for fraud detection due to this trade-off [18]
The Neural Network model exhibits outstanding performance with a high True Positive Rate of 98.78% and a very low False Positive Rate of 0.62%, resulting in an excellent accuracy of 99.38%. However, the computational speed of 3256.32 seconds is significantly slower compared to other models, making it less suitable for scenarios requiring real-time or near-real-time processing. Despite its accuracy, the model's slow processing time could limit its practical application in time-sensitive environments [18]
k-Nearest Neighbors (k-NN) demonstrated good performance in fraud detection, with a True Positive Rate of 82.55% and a False Positive Rate of 0.44%. This low false positive rate ensures that few legitimate transactions are wrongly flagged as fraudulent. Moreover, the model achieved a high accuracy of 99.54%, indicating its overall reliability in detecting fraud. The computational speed of 118.14 seconds is reasonable, but slower than Logistic Regression or XGBoost, making it less practical for real-time applications. Additionally, k-NN is known for being computationally intensive as it performs distance calculations on the entire training set during prediction. Therefore, while k-NN provides good results, its scalability and real-time applicability are limited compared to other models.
Random Forest achieved the highest accuracy of all the models, at 99.92%, and an excel-

lent True Positive Rate of 96.19%, meaning it correctly identifies nearly all fraudulent transactions. Additionally, the False Positive Rate is the lowest among all models, at just 0.07%, indicating that very few legitimate transactions are wrongly flagged. However, Random Forest's major disadvantage is its computational speed, with a processing time of 4682.67 seconds. This makes it the slowest model, significantly limiting its use in real-time fraud detection systems. While Random Forest offers the best performance in terms of accuracy and minimizing false positives, its computational inefficiency suggests that it would need optimization or hardware acceleration to be practical in time-sensitive environment [23] [18].
XGBoost emerged as one of the top-performing models, with a True Positive Rate of 98.99% and a low False Positive Rate of 0.25%. This results in a high accuracy of 99.75%, making XGBoost a reliable option for fraud detection. The model's computational speed of 99.08 seconds is also highly favorable, placing it among the fastest models. XGBoost provides an excellent balance of high detection accuracy and fast processing times, making it one of the best choices for real-time fraud detection systems. Its ability to handle large datasets efficiently, combined with its high performance, makes XGBoost a standout model in this evaluation. [20].

Considering both detection performance and computational speed, XGBoost and Logistic Regression emerge as strong candidates for fraud detection. XGBoost offers a high accuracy (99.75%) and relatively fast computational speed (99.08 seconds), making it well-suited for real-time environments. Logistic Regression, while slightly less accurate (94.92%), provides the fastest computational speed (61.79 seconds), making it ideal for systems that prioritize both reliability and quick responses.

Models like Neural Networks and Random Forest, despite their high accuracy and low false positive rates, are limited by their slow processing times, which hinders their practicality for real-time fraud detection systems. k-NN, although offering high accuracy (99.54%) and a low false positive rate, also struggles with scalability and real-time performance due to its relatively slower processing time (118.14 seconds).

Therefore, for real-time fraud detection systems, XGBoost and Logistic Regression are recommended due to their balance between speed and accuracy. Neural Networks and Random Forest could be considered in environments where detection accuracy is the primary concern and speed is less critical, with hardware optimizations or faster

processing needed to improve their performance. k-NN may be viable in smaller-scale systems but may require further tuning or hybrid approaches to be efficient in larger, real-time applications.

CHAPTER 6

Conclusions

The primary aim of this dissertation was to evaluate various anomaly detection techniques for detecting fraudulent transactions in mobile money systems. Through the application of supervised machine learning models, this research sought to identify the most effective approaches for real-time fraud detection, balancing detection accuracy with computational efficiency.

The findings of this study indicate that models such as XGBoost and Logistic Regression are highly suitable for mobile money fraud detection. XGBoost demonstrated an excellent balance between accuracy (99.75%) and computational speed (99.08s), making it an ideal candidate for real-time deployment. Logistic Regression, while slightly less accurate (94.92%), provided the fastest computational speed (61.79s), making it particularly suited for systems that prioritize both speed and reliability in fraud detection.

On the other hand, methods like Neural Networks and Random Forest, while showing near-perfect true positive rates (98.78% and 96.19%, respectively), were limited by their slow computational speeds (3256.32s and 4682.67s, respectively). This restricts their applicability in real-time systems unless hardware optimizations are applied. k-Nearest Neighbors (k-NN) performed well in terms of accuracy (99.54%) and false positive rate (0.44%), but its computational speed (118.14s) may hinder its use in larger-scale real-time applications. Naive Bayes, despite being computationally efficient (4.19s), had a very high false positive rate (43.44%), making it unsuitable for practical fraud detection systems where accuracy is critical.

Despite the positive results, this research faced certain limitations. The use of

synthetic data, while useful for simulating real-world scenarios, may not capture the full complexity and unpredictability of actual fraud patterns in live mobile money systems. Further testing with real-world datasets is necessary to validate these findings. Moreover, the high imbalance in fraud data, which necessitated the use of SMOTE for supervised models, remains a challenge in ensuring consistent model performance across different datasets.

To further enhance the performance of fraud detection systems, future research could explore hybrid approaches, combining the strengths of different models to improve accuracy and reduce false positives. Additionally, optimizing the computational performance of resource-intensive models like Neural Networks and Random Forest could make them more feasible for real-time applications.

In conclusion, this dissertation has demonstrated the potential of machine learning models in detecting fraudulent transactions within mobile money systems. As mobile financial services continue to expand globally, the need for robust, scalable fraud detection systems becomes more critical. By refining and optimizing these models, we can contribute to a safer, more secure financial ecosystem, protecting both users and businesses from the ever-evolving threat of fraud.

APPENDIX

Python Code for all visualizations and models used

```
# ----- A SCATTER PLOT WITH A CLEAR OUTLIER POINT ----------

import numpy as np
import matplotlib.pyplot as plt

# Generating random data
np.random.seed(567)
x = np.arange(1, 101)
y = x ** 2 + np.random.normal(0, 500, size=x.size)

# Introduce an outlier
y[40] = 9000  # Add a significant outlier

# Create scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(x, y, color='red')
```

```
# Add labels and title
plt.title('Scatter Plot with Outlier')
plt.xlabel('Transaction Index')
plt.ylabel('Transaction Value')

# Display the plot
plt.savefig('outlier_in_scatter.png')
plt.show()

# ----- EXPLORING GSMA State of Mobile Money Dataset ------

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Read the dataset and display the first 10 records
mm = pd.read_excel("Global_Mobile_Money_Dataset_2023.xlsx",
sheet_name="All Data Table")
print("First 10 records of the dataset:")
print(mm.head(10))

# Fill null values with zeros
mm.fillna(0, inplace=True)

# Check unique values in the 'Unit' column
print("\nUnique values in 'Unit' column:")
print(mm['Unit'].unique())

# Apply filters to get the relevant data
filtered_mm = mm[
    (mm['Geo_view'] == 'Region') &
```

```
    (~mm['Geo_name'].isin(['Africa',
    'Global'])) &  #No 'Africa' and 'Global'
    (mm['Attribute'] == 'Registered') &
    (mm['Unit'] == 'Accounts') &
    (mm['Metric'] == 'Mobile Money accounts') &
    (mm['Measure'] == 'Registered Accounts')
]

# Display the filtered DataFrame
print("\nFiltered DataFrame:")
print(filtered_mm.head())

## Mobile Money Uptake By Region in 2023

# Select only the data for the year 2023
region_data = filtered_mm[['Geo_name', '01/12/2023']]
region_data.columns = ['Region', 'Total Accounts']

# Plotting the data
plt.figure(figsize=(10, 4))
sns.barplot(data=region_data, x='Region',
y='Total Accounts', palette='viridis')
plt.title('Mobile Money Uptake by Region in 2023')
plt.xlabel('Region')
plt.ylabel('Number of Accounts')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('Regions_summary.png')
plt.show()

## Growth of Mobile Money Accounts Since 2001
```

```
date_columns = filtered_mm.columns[6::4]

# Aggregate and plot the trend of mobile money accounts
filtered_trend_data = filtered_mm[date_columns].sum()

# Plot the aggregated data over time
plt.figure(figsize=(12, 6))
plt.plot(filtered_trend_data.index,
filtered_trend_data.values, marker='o', color='blue')
plt.title('Trend␣of␣Registered␣Mobile␣Money
Accounts␣Over␣Time')
plt.xlabel('Time')
plt.ylabel('Number␣of␣Registered␣Mobile␣Money␣Accounts')
plt.xticks(rotation=90)
plt.grid(True)
plt.tight_layout()
plt.savefig('mm_trend.png')
plt.show()

# --- EXPLORATORY DATA ANALYSIS ON THE MOBILE MONEY DATASET----

import seaborn as sns

# Read the dataset

data = pd.read_csv("PS_20174392719_1491204439457_log.csv")
data.head(20)

data.info()
data.shape
data.describe().T
```

```
print("Missing values: ",data.isnull().sum().sum())

# ---THE TYPE OF TRANSACTION USED THE MOST ----------

# Count the number of transactions per type
transaction_counts = data['type'].value_counts()

# Plot the transaction counts
plt.figure(figsize=(10, 6))
sns.barplot(x=transaction_counts.index,
y=transaction_counts.values, palette="viridis")
plt.title('Transaction Type Counts')
plt.xlabel('Transaction Type')
plt.ylabel('Count')
plt.show()

# ------ DISTRIBUTION OF TRANSACTION AMOUNTS ----------

# Define the bins (ranges) for the buckets
bins = [0, 100000, 200000,
500000, 1000000, data['amount'].max()]

# Create labels for these bins
labels = ['0-100,000', '100,001-200,000',
          '200,001-500,000', '500,001-1,000,000',
           f'1,000,001-{int(data["amount"].max()):,}']

# Cut the data into these bins
data['amount_bucket'] = pd.cut(data['amount'],
bins=bins, labels=labels, include_lowest=True)

# Count the number of transactions in each bucket
```

```
bucket_counts = data['amount_bucket'].
value_counts().sort_index()

# Convert to a DataFrame for better display
bucket_distribution = pd.DataFrame({'Transaction␣Amount':
bucket_counts.index, 'Count': bucket_counts.values})


bucket_distribution_styled = bucket_distribution.style.
format({'Count': '{:,}'}).
background_gradient(subset=['Count'], cmap='viridis')

# Display the table with formatting
bucket_distribution_styled

# ----------CORRELATION MATRIX --------------

# Select only numeric columns for correlation matrix
numeric_df = data.select_dtypes(include=[float, int])

# Compute the correlation matrix
corr_matrix = numeric_df.corr()

# Plot the heatmap of the correlation matrix
plt.figure(figsize=(14, 7))
sns.heatmap(corr_matrix, annot=True, cmap='viridis', fmt='.2f')
plt.title('Correlation␣Matrix')
plt.savefig('correlation_nmatrix.png',bbox_inches='tight')
plt.show()

# -----DISTRIBUTION OF FRAUDULENT TRANSACTIONS -------
```

```
fraud_counts = data['isFraud'].value_counts().reset_index()

# Rename the columns for better readability
fraud_counts.columns = ['isFraud', 'Count']

# Display the table
from IPython.display import display
display(fraud_counts)

# ----DAILY TRANSACTION AMOUNTS ----------

# Separate the data into legitimate and
fraudulent transactions
legitimate = data[data['isFraud'] == 0]
fraudulent = data[data['isFraud'] == 1]

plt.figure(figsize=(12, 6))
plt.plot(data['step'], data['amount'], color="blue")
plt.title('All Transactions - Amount')
plt.xlabel('Time step')
plt.ylabel('Amount of transactions')
plt.savefig('all_transactions.png',bbox_inches='tight')
plt.show()

plt.figure(figsize=(12, 6))
plt.plot(fraudulent['step'], fraudulent['amount'], color='red')
plt.title('Fraudulent Transactions - Amount')
plt.xlabel('Time step')
plt.ylabel('Amount of transactions')
plt.savefig('fraudulent_transactions.png',bbox_inches='tight')
plt.show()
```

```
plt.figure(figsize=(12, 6))
plt.plot(legitimate['step'], legitimate['amount'], alpha=0.6)
plt.title('Legitimate Transactions - Amount')
plt.xlabel('Time step')
plt.ylabel('Amount of transactions')
plt.savefig('legitimate_transactions.png',bbox_inches='tight')
plt.show()

# ----DATA PRE PROCESSING AND MODELLING ------------

import numpy as np
import time
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE
from sklearn.metrics import confusion_matrix, accuracy_score
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv('PS_20174392719_1491204439457_log.csv')

# Define features and labels
X = df.drop(['isFraud', 'isFlaggedFraud',
'nameOrig', 'nameDest'], axis=1)
y = df['isFraud']

# Encode categorical features
X = pd.get_dummies(X, columns=['type'])

# Split the dataset into training and testing sets
```

```
X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.3, random_state=42, stratify=y)

# Standardize the numeric features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Apply SMOTE to balance the dataset
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote =
smote.fit_resample(X_train, y_train)

# Define a function to calculate the required metrics
def calculate_metrics(y_test, y_pred):
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
    accuracy = accuracy_score(y_test, y_pred)
    true_positive_percentage = tp / (tp + fn)
    false_positive_percentage = fp / (fp + tn)
    return accuracy, true_positive_percentage,
            false_positive_percentage

# Train and Test the models
def train_and_evaluate_model(model, model_name):
    # Record the start time
    start_time = time.time()

    # Train the model
    model.fit(X_train_smote, y_train_smote)

    # Record the end time
    end_time = time.time()
```

```
    # Calculate the time taken
    time_taken = end_time - start_time

    # Make predictions
    y_pred = model.predict(X_test)

    # Calculate metrics
    accuracy, tpp, fpp = calculate_metrics(y_test, y_pred)

    # Print the results
    print(f"{model_name}\n")
    print(f"Time Taken: {time_taken:.4f} seconds")
    print(f"Accuracy: {accuracy:.4f}")
    print(f"True Positive Percentage: {tpp:.4f}")
    print(f"False Positive Percentage: {fpp:.4f}\n")
    print("Confusion Matrix:")
    print(confusion_matrix(y_test, y_pred))
    print("\n")

# Define the models
from sklearn.neighbors import KNeighborsClassifier
#from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neural_network import MLPClassifier

models = [
    (KNeighborsClassifier(n_neighbors=5), "k-NN Classifier"),
    #(SVC(kernel='rbf', random_state=42), "SVM Classifier"),
```

```
    (RandomForestClassifier(n_estimators=100, random_state=42),
    "Random Forest Classifier"),
    (XGBClassifier(random_state=42, use_label_encoder=False,
    eval_metric='logloss'), "XGBoost Classifier"),
    (LogisticRegression(random_state=42),
    "Logistic Regression Classifier"),
    (GaussianNB(), "Naive Bayes Classifier"),
    (MLPClassifier(random_state=42),
    "Neural Network Classifier")
]

# Train and evaluate each model
for model, model_name in models:
    train_and_evaluate_model(model, model_name)
```

Bibliography

[1] M. E. Lokanan. Predicting mobile money transaction fraud using machine learning algorithms. *Journal of Financial Crime*, 12 July 2023. https://doi.org/10.1002/ail2.85.

[2] T. Suri. Mobile Money. *Annual Review of Economics*, vol. 9, pp. 497-520, 2017. https://doi.org/10.1146/annurev-economics-063016-103638.

[3] GSMA. State of the Industry Report on Mobile Money. *Annual Review of Economics*, 2024. https://www.gsma.com/sotir/.

[4] Economic Times. 5 reasons why consumers still don't use digital payments. *Economic Times*, June 22, 2018. https://economictimes.indiatimes.com/wealth/spend/5-reasons-why-consumers-still-dont-use-digital-payments/articleshow/64699938.cms.

[5] World Bank. Financial Inclusion Overview. *World Bank*, 2024. https://www.worldbank.org/en/topic/financialinclusion/overview

[6] K. Martin. Anomaly Detection: A Survey *Communications of the ACM*, 52(12):52-56, 2009. https://dl.acm.org/doi/pdf/10.1145/1541880.1541882.

[7] Cuemath. What is an Outlier in Math? *Cuemath*, 2024. https://www.cuemath.com/data/outlier/.

[8] Wolfram MathWorld. Outlier. *Wolfram MathWorld*, 2024. https://mathworld.wolfram.com/Outlier.html.

[9] Encyclopedia of Mathematics. Outlier. *Encyclopedia of Mathematics*, 2024. https://encyclopediaofmath.org/index.php?title=Outlier.

[10] A. Reurink Financial Fraud. A Literature review *Journal of Economic Surveys*, 32(3):1-14, 2018. https://onlinelibrary.wiley.com/doi/epdf/10.1111/joes.12294.

[11] TechCabal. Botswana Ecoplexus Scam. *TechCabal*, May 13, 2023. https://techcabal.com/2023/05/13/botswana-ecoplexus-scam/.

[12] Botswana Labour Market Observatory. Labour Statistics: Unemployment rate is 25.4 percent. *Botswana Labour Market Observatory*, 2024. https://www.botswanalmo.org.bw/labour_statistics#:~:text=Unemployment%20rate%20is%2025.4%20percent%20%28Quarterly,Multi-topic.

[13] Impact of Fraud The Impact of Fraud. *Financial Crime Academy*, 2024. https://financialcrimeacademy.org/impact-of-fraud/#:~:text=Fraud%20can%20distort%20markets%20by%20giving%20fraudsters%20a,community%20services%20that%20assist%20those%20affected%20by%20fraud..

[14] Techopedia. Credit Card Fraud Statistics. *Techopedia*, 2024. https://www.techopedia.com/credit-card-fraud-statistics.

[15] Anti Money Laundering Updated List of at Risk Third Country- Jurisdictions *Anti Money Laundering*, 2022. https://finance.ec.europa.eu/news/anti-money-laundering-commission-updates-its-list-high-risk-third-co en.

[16] E. A. Lopez-Rojas, A. Elmir, and S. Axelsson. PAYSIM: A Financial Mobile Money Simulator for Fraud Detection. 2016.

[17] Z. Seng. SMOTE: Synthetic Minority Over-sampling Technique. *Medium*, 2023. https://medium.com/@ziad.seng/smote-synthetic-minority-over-sampling-technique-3eb26ddd7aec#:~:text=SMOTE%20is%20a%20powerful%20technique%20for%20addressing%20the,learning%20models%20perform%20better%20on%20minority%20class%20predictions.

[18] J. O. Sinayobye, F. Kiwanuka, and S. K. Kyanda. A State-of-the-Art Review of Machine Learning Techniques for Fraud Detection Research. In *Proceedings of the ACM IEEE Symposium on Software Engineering in Africa (SEIA 18)*, May 27, 2018, Gothenburg, Sweden. ACM, 2018. https://doi.org/10.1145/3195528.3195534.

[19] G. James, D. Witten, T. Hastie, and R. Tibshirani. *Introduction to Statistical Learning with Python* Springer, 2021

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

[21] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. 4th ed., Morgan Kaufmann, 2017.

[22] A. Toshniwal, K. Mahesh, and J. R. Overview of Anomaly Detection Techniques in Machine Learning. In *Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, IEEE, 2020. https://doi.org/10.1109/I-SMAC.2020.5464640.

[23] S. B. Edwin Raj and A. A. Portia. Analysis on Credit Card Fraud Detection Methods. In *Proceedings of the International Conference on Computer, Communication and Electrical Technology (ICCCET 2011)*, IEEE, March 2011, pp. 152-156. https://doi.org/10.1109/ICCCET.2011.5762459.