Open Peer Review on Qeios

Predicting Mobile Money Transaction Fraud using Machine Learning Algorithms

Mark Lokanan[1]

1 Royal Roads University

Funding: The author(s) received no specific funding for this work.

Potential competing interests: The author(s) declared that no potential competing interests exist.

Abstract

The ease with which mobile money is used to facilitate cross-border payments presents a global threat to law enforcement in the fight against laundering and terrorist financing. This paper aims to use machine learning classifiers to predict transactions flagged as fraud in mobile money transfers. Data for this paper came from real-time transactions that stimulate a well-known mobile transfer fraud scheme. This paper uses logistic regression as the baseline model and compares it with ensembles and gradient descent models. The results indicate that the established logistic regression model did not perform too poorly compared to the other models. The random forest classifier had the most outstanding performance among all measures. The amount of money transferred was the top feature to predict money laundering transactions through mobile money transfers. These findings suggest that more research is needed to improve the logistic regression model. The random forest classifier should be further explored as a potential tool for law enforcement and financial institutions to detect money laundering activities in mobile money transfers.

Mark Lokanan[1]

[1] *Faculty of Management | Royal Roads University 2005 Sooke Road, Victoria, BC Canada V9B 5Y2 | royalroads.ca*

1. Introduction

With the increasing popularity of mobile money services, there has been a corresponding increase in fraud and money laundering cases. Mobile money providers must therefore be vigilant in combating such activity. One way to combat fraud is to require users to provide additional information when making transactions, such as a PIN or biometric data. Money laundering cases can be more difficult to detect, but mobile money providers can look for patterns of suspicious activity, such as unusually large or frequent transactions. The Financial Action Task Force (FATF) noted that mobile money payment presents a global threat to money laundering and terrorist financing as it can be used to facilitate cross-border payments without the need for a bank account. In response, the FATF released risk-based approaches to countering the threat. The guidance sets out recommendations for identifying and managing risk, including the use of computational technology.

The use of machine learning (ML) and artificial intelligence (AI) is increasingly seen to combat mobile money fraud and address anti-money laundering (AML) compliance. Computational technology has always played a role in the fight against financial crime, but the rise of ML and AI is giving law enforcement a powerful new tool in the battle against mobile money fraud. AI can help financial institutions to identify and flag suspicious behaviour, such as large or unusual transactions, and better understand their customers' needs and risk profiles. By harnessing the power of AI, financial institutions can significantly improve their ability to combat mobile money fraud and address money laundering threats. This paper aims to use ML learning algorithms to build a fraud detection model that will detect red flags of fraud and money laundering from mobile money transactions. More specifically, this paper will use a set of risk-based indicators to predict how likely a transaction will be fraudulent.

This study provides several significant advances to the existing body of research on methods for detecting suspicious transactions in mobile money transfers. In theory, machine learning algorithms can circumvent the challenges of attempting to identify illegal transactions by relying on the more conventional rule-based benchmark methodology. In the classic rule-based benchmark technique, identifying illegal transactions is accomplished by using predefined criteria based on mathematical conditions. The rule-based approach is time-consuming and costly and has a high rate of false-positive results. ML addresses these issues by enabling computers to learn from the data and make predictions. When applied to mobile money, ML can be used to enable automated detection of potentially fraudulent transactions. An example of this would be training a ML algorithm on a dataset containing transactions that are known to be fraudulent. Based on this learned experience, the algorithm can be tuned to find future fraudulent transactions by looking for patterns similar to those found in the training data.

Practically, we propose a novel data-driven method of fraud detection that has been precisely tuned to the distinctive features of mobile money transactions. This strategy uses ML to automatically identify suspicious transactions in real time, eliminating the need for extensive human involvement. The ML approach, which uses real-time analysis, can quickly spot transactions that could be fraudulent and stop them from going through and reduce the number of fraudulent transactions.

The remainder of this paper is structured in the following manner. Section two thoroughly analyzes the literature on ML and mobile money transfers concerning fraud and money laundering. Section three discusses the methodology and algorithms considered for the ML models. Section four examines and analyzes the results. Section five concludes with limitations in ML for fraud research and identifies opportunities for further study.

2. Literature Review on Mobile Money Fraud and Money Laundering

Mobile Money Services (MMS) or Mobile Money Transfer Services (MMTS) are unbanked financial services that operate primarily through smartphone apps supported by mobile operators or banking institutions and are frequently referred to as branchless banking services for their users [1][2]. They facilitate the fund transfer of electronic cash using the users' mobile phones while not involving any bank account in the process[3][4]. A few common examples of MMS or MMT services are Tigopesa, M-Pesa, Simbanking, and NMB Mobile, offered by Tigo Tanzania Ltd, Vodacom Tanzania, CRDB Bank, and National Microfinance Bank respectively[5]. Ideally, MMT enables person-to-person (P2P) payments for the customers, and the services supported by the mobile money system involve participation from various stakeholders like mobile users, regulators, mobile network operators, telecom retailers, agents, and financial institutions[6]. The mobile users act as the customers for the MMT services, while mobile network operators (MNOs) completely facilitate the ecosystem of MMS in conjunction with telecom retailers and agents, who are responsible for opening accounts for the customers, conducting customer due diligence, and other compliance activities like KYC - Know Your Customer. Financial institutions and regulators assist MNOs in establishing financial inclusion and risk management mechanisms, whereas MNOs limit banks to processing payment delivery, clearing, and settlement[6]. A bank can or cannot be involved in the MMS depending on the adopted model of MMT[7]. These players collectively enable MNOs to implement the new P2P payment facility for unbanked users.

The number of users using mobile money for small or large transactions has increased drastically in the last decade[1]. Research estimates that this number is expected to rise with the increasing dependency and usage of mobile phones in the future[8][9]. Due to their success and popularity, mobile money systems are set to attract the attention of fraudsters interested in laundering the proceeds of crime[10][11][12][13]. Fraudsters can launder money by seizing the details related to several mobile money transfers during transmission or creation and saving the server's data through phishing attacks or viruses, which can then be misused to launder illicit funds[8][14]. Similarly, the reprobate end users of MMS can launder their dirty money through this system by smurfing a large chunk of the illegitimate source of income into a small number of mobile money transactions, using multiple accounts and phones while avoiding the suspicious nature of the act[3][6]. Indeed, some speculate that this system could be used to fund terrorist activities, though there is evidence that launderers have used mobile transfers to launder funds for terrorist financing[13]. These findings have brought the need for more advanced technology to identify and control the risks associated with the mobile money system.

2.1. Detecting Mobile Money Fraud and ML Using Computational Technology

Technological innovation can be useful in mitigating various risks associated with the MMS. Improving technological surveillance by increasing the security, resilience, and scalability of MNO networks used in MMT can reduce risks associated with mobile money fraud (MMF) to some extent at the security and procedural levels[15]. Implementing the two-factor authentication model for securing communications through SMS in MMT has proven very effective[16]. The most important contribution technology can make, other than the new developments in the security information and events management field, is through innovation or by designing MMF and money laundering prediction or detection tools for the MMS. In the following paragraphs, we will focus on using ML algorithms and artificial intelligence to mitigate the risk of MMF and money laundering in mobile money transactions.

2.2. Machine Learning and Artificially Intelligent Algorithms

Technology is pivotal in investigating and detecting fraudulent or laundered mobile money transactions[17]. ML, AI, and data mining have proven effective in detecting MMF and money laundering activities in the MMS[18][19]. More specifically, ML algorithms teach computers to learn human behaviour and detect patterns in the data[19][20]. Supervised ML algorithms like logistic regression, decision tree, gradient descent, and random forest have all been successfully used in detecting financial fraud from labelled data[18][20][21][22][23][24]. The following sections are devoted to reviewing the literature on these algorithms.

2.2.1. Logistic Regression

Logistic regression will be used as the baseline algorithm to compare with the other models. Logistic regression uses a linear combination of input variables (x) to predict an output variable (y)[22]. The output variable is usually (0 or 1), representing the two possible outcomes of a binary classification task (e.g., fraud or not fraud). The coefficients of the input variables (β) are estimated using maximum likelihood estimation. The Sigmoid function is a mathematical function that is the foundation of logistic regression and takes an actual number and translates it into a value between 0 and 1. The Sigmoid translation is important for ML learning classification tasks because it allows the algorithm to easily separate data points into different classes[25]. The sigmoid function is denoted in equation 1.

Where

$f(x)$ is the value bounded between 0 and 1,

x is the derivative of the sigmoid function,

e is the mathematical constant

$$f(x) = 1 \;/\; (1 + e^{-x}) \qquad \text{eq. 1}$$

The output of the sigmoid function can be interpreted as a probability. For example, if the output of the Sigmoid function is 0.8, this can be interpreted as an 80% chance that the data point belongs to one class and a 20% chance that the data point belongs to the other class[26][27]. Using the Sigmoid function, a logistic regression model can be trained to predict the class to which a new data point belongs. The logistic regression classifier uses the Sigmoid function to estimate the probability that $y = 1$, given the size of$x$. Equation 2 denotes the logistic regression model.

Where

$Y =$ values between 0 and 1,

$e\beta_{0+}\beta_{1+}X$ represents the independent features, and

β_0 and β_1 will give different estimations of Pr

$$Pr(Y = \mathbf{1} \mid X = x) = \frac{e\beta_{0+}\beta_{1+}X}{1 + e\beta_{0+}\beta_{1+}X} \qquad \text{eq. 2}$$

Logistic regression is a valuable technique for fraud classification tasks[22][26][27]. Research has shown that the logistic regression performed relatively well and, in some cases, outperformed other classifiers in fraud classification tasks[22][25][27][28]. Logistic regression has been used in a variety of domains to predict fraud. For example, logistic regression has been used in the financial sector to detect credit card and insurance fraud[25][29]. Others have used logistic regression to predict medical billing fraud with reliable results[30][31]. Logistic regression models have several advantages over traditional fraud detection methods. First, it is highly scalable and can be applied to large data sets[31]. Second, it is highly effective at detecting fraud, with a success rate that is generally much higher than traditional methods[25]. In addition, logistic regression models are relatively easy to interpret, which makes them valuable tools for fraud analysts[22][27]. Finally, it is relatively easy to deploy and use in production systems[22][25]. However, logistic regression is not without drawbacks; in particular, it can be susceptible to overfitting if the data is not carefully preprocessed[26][29]. Even though model overfitting is a problem, logistic regression is an excellent way to build a fraud detection model that can be used as a benchmark to compare with other classifiers.

2.2.2. Decision Tree

Another useful machine learning algorithm for fraud detection is the decision tree classifier[21]. Decision tree employs a tree structure for choice making, where the root symbolizes the fundamental decision, edges display the decision node, leaves show the class labels that convey the decision, and internal modes indicate qualities picked based on information gain or Gini Index[20][32]. Typically considered a weak learner, decision tree classification ability is boosted by using the gradient boosting technique[33]. Gradient boosting is an ensemble learning technique that optimizes performance accuracy by sequentially generating the decision tree so that it is always superior to the previous one.[18][20]. This project employs the Gini Index to label the data. The mathematical formula for Gini Index is shown in equation 2:

Where

fk is the fraction of items labeled with$k$ in the set and $\sum fk = 1$.

$$I_G(f) = \sum_{k=1}^{m} f_k(1 - f_k) \qquad \text{eq. 4}$$

Concerning fraud detection, a decision tree involves building a model that can predict whether an observation is legitimately derived or not[34]. The decision tree model is based on a series of yes-or-no questions, each narrowing down the possible outcomes (i.e., fraud or no fraud)[35]. For example, a decision tree for fraud detection might ask whether the transaction is consistent with the customer's past behaviour. If the answer is no, it could be flagged as potentially fraudulent. Once the model is built, it can be used to classify new data points as either fraudulent or non-fraudulent. Decision tree algorithms are highly effective in identifying fraud. They are often used with other methods, such as rule-based systems and ML[20][21][35]. Classification algorithms based on decision trees are a powerful way to find fraud because they can help find even the most complex kinds of fraud.

2.2.3. Gradient Descent

Gradient descent is a machine learning algorithm that uses first-order iterative optimization to find the minimum of a function. To locate a function's local minimum using gradient descent, one must take steps proportional to the function's negative gradient (or approximate gradient) at the current point[20]. Instead, if one takes steps proportional to the gradient's positive, one approaches a local maximum of that function, known as gradient ascent[36]. It is an optimization algorithm used to find the values of parameters (coefficients) of a function (f) that minimizes a cost function (c). The cost function is a measure of how far away the predicted values are from the actual values. The algorithm iteratively adjusts the coefficients until it converges on a set of coefficients that minimizes the cost function[36][37].

The algorithm is represented by the probabilistic formula where the likelihood function$p(x, 0, 1)$ predicts the probability of a binary outcome given a set of independent variables. In this case, the algorithm is trained to predict whether an instance belongs to class 0 or 1, which are represented by the labels = 0 and = 1. The coefficients 0 and 1 represent the probability of the output y to be 1 or 0 given x. In other words, 0 and 1 are the log odds of the output being 1 or 0 given x [20][36]. The gradient descent algorithm is popular for machine learning applications, particularly in fraud detection, because the algorithm can learn from data very quickly and effectively. Additionally, the gradient descent algorithm can handle very large datasets, making it ideal for fraud detection applications requiring high accuracy. Numerous studies have been conducted on the efficacy of the gradient descent algorithm for fraud detection. Most of these studies have found that the algorithm effectively detects fraudulent activities[38][39]. Recent studies have found that the algorithm was very effective in detecting known fraud cases in a dataset of credit card transactions[38][40]. Another study found that the algorithm could successfully identify fraudulent insurance claims with high accuracy[39]. Furthermore, the gradient descent algorithm is also relatively easy to implement, which makes it a good choice for organizations that do not have a lot of resources or expertise in fraud detection methods.

2.2.4. Random Forest

Random Forest is a machine learning algorithm for classification and regression tasks and is learned base on the decision tree concept[24][25][41]. The random forest algorithm generates a series of decision trees, each created using a randomly chosen subset of the training data[42][43]. Even though random forest is a useful learning algorithm that can be used to solve both linear and nonlinear problems, it is especially useful for addressing nonlinear data[18][41]. The predictions of the individual trees are then combined to produce the final prediction. The random forest algorithm is effective because it reduces the prediction variance while maintaining the model's accuracy[43][44]. Additionally, the random forest algorithm, when pruned, is resistant to overfitting, which means it can handle large datasets and generalize well to new data. Because of these advantages, the random forest algorithm is a powerful tool for machine learning applications.

The advantage of the random forest algorithm for classification tasks is that it can help reduce the number of false positives generated by other AI methods, such as neural networks[42]. In addition, the random forest technique is not difficult to construct, and it is possible to execute the algorithm on large datasets with accurate performance[44]. These two features combine to make the algorithm a powerful instrument for discovering patterns in data. In particular, the random forest classifier is well-suited for detecting fraud, often identifying unusual patterns in the data[21][43]. For example, fraudsters might create multiple accounts with different email addresses and use them to make small purchases to avoid detection. Alternatively, they might try to return items they never bought to receive a refund. By looking for these and other unusual patterns, the random forest algorithm can help to detect fraud before it results in significant losses. In addition, the random Forest classifier has also been used in other domains such as loan default, credit risks, image recognition, and medical diagnosis[21][25][41][43]. The random forest classifier has proven to be a versatile tool algorithm and has been used in various domains.

3. Research Design and Experimental Setting

3.1. Data Generation and Stimulation

Currently, there is a lack of data on fraud and money laundering detection[45]. One of the reasons cited for this outcome is confidentiality and the sensitivity of the data. Researchers have developed stimulators that use algorithms to generate synthetic data from real-time observations to address this problem. Some of the most prominent stimulators used by researchers are the Mobile Money Simulator (PaySim) and Retail Store Simulator (RetSim)[45][46]. These simulators allow researchers to generate synthetic transactional data that contains both legitimate and fraudulent transactions.[45] and [46] demonstrated using Agent-Based Simulation (ABS) and Multi Agent-Based Simulations (MABS) that synthetic transactional data developed by PaySim and RetSim are as useful as real transaction data for detecting MMF and money laundering activities while retaining the reliability and confidentiality of the actual transaction data.

The data for this project came from an MABS that was used to calibrate real-time transactions. The data came from Lopez-Rojas and his colleagues' work, who use MABS to develop agents representing clients and merchants in PaySim and customers and salesmen in RetSim [45][46]. The data is simulated and uses a real-world scenario based on a well-known fraud scheme to demonstrate the superiority of simulated data over real-world data when establishing adequate controls for fraud detection[47]. The usual behaviour was derived from the behaviour that was observed in the data collected. This behaviour is enshrined in the agents' rules governing the transactions and interactions between consumers and salespeople or between customers and merchants. Based on patterns of actual fraud[45], some of these agents were set up to commit fraud.

3.2. Data Description and Variables

PaySim is used to simulate mobile money transactions in this dataset. The simulations are based on a sample of actual mobile money transactions that were taken from one month's worth of financial logs generated by a mobile money service that was deployed in an African nation. The first logs were given by a global firm that is the supplier of the mobile financial service presently operational in more than 14 countries. The company provided the original logs. In total, 1048575 rows of data were collected that comprised nine independent features. Table 1 depicts the features and target variables that represent the dataset.

Table 1: Independent Features and Description

Features	Description	Measures	Indicators
Step	Map the unit of time	Continuous	1 step = 1 hour
Type	Type of transfer	Categorical	Cash-in, Cash-out, Debit, Payment, Transfer
Amount	Amount of the transaction	Continuous	
nameOrig	Customer who started the transaction	Continuous	
oldbalanceOrg	Initial balance before transaction	Continuous	
newbalanceOrg	New balance after transaction	Continuous	
nameDest	Recipient of transaction	Continuous	
oldbalanceDest	Initial balance recipient before the transaction	Continuous	
newbalanceDest	New balance recipient after the transaction	Continuous	

The dependent variable is fraud. "Fraud," in this context, refers to the transactions carried out by fraudulent actors inside the simulation. More specifically, the fraudulent activity of the agents tries to profit by seizing control of client accounts and laundering the cash by moving them to another system. The funds are then cashed out of the system. Fraud was coded as 1 = fraud and 0 = no fraud and is represented in equation 1.

$$y = \{1, fraud, 0 no - fraud\} \qquad \text{eq. 1}$$

3.3. Data Cleaning and Preprocessing

Some features were redundant and had to be dropped before model building. As a result, the features "nameorig" and "nameDest" are no longer relevant and must be removed. There was no longitude or latitude associated with these features to locate the destinations. There was no variation in the feature "isFlaggedFraud," and it was also dropped from the model.

3.3.1. Feature Scaling

One of the most important aspects of preprocessing data for ML is feature scaling. Preprocessing is especially necessary when the features are on different scales and span a wide range of values. ML models are highly sensitive to features with different scales and, if not handled properly, can throw off the model and lead to sub-optimal performance or even incorrect predictions. There are a few different ways to scale features, but the most common is min-max scaling.

This approach scales all values to be between 0 and 1. Other methods include standardization, which scales values so that they have a mean of 0 and a standard deviation of 1. The data for this project contains features with different scales and ranges. Given that the data was not normally distributed, normalization with MinMaxScaler was used to normalize the data. The formula used to normalize the data is shown in equation 5.

$$\frac{X - \bar{X}}{X - X}_{Min\ Max}\{0 \longrightarrow 1\} \text{ eq. } 5$$

3.3.2. SMOTe-ENN for Imbalance Data

The dataset used for this study was highly imbalanced. The fraud to no fraud ratio was 99% (no fraud) and 1% (fraud). One common approach when working with imbalanced datasets is to upsample the minority class. Upsampling can be done in various ways, but one popular method is the Synthetic Minority Oversampling Technique (SMOTe) plus Edited Nearest Neighbour (ENN). The SMOTe-ENN method combines the SMOTe and ENN algorithms to improve the performance of the ML classifiers[48][49]. SMOTe creates synthetic minority examples by interpolating between existing minority examples[50]. ENN then cleans up the resulting oversampled data set by removing outliers[51]. The SMOTe-ENN method has been shown to be more effective than either algorithm alone[48][49][51]. SMOTE-ENN is particularly effective at handling imbalanced data sets, often in real-world applications. As a result, the SMOTe-ENN method is often used in fields such as credit scoring and fraud detection[49][51].

Figure 1 shows the data before SMOTe-ENN resampling. Because of the imbalanced nature of the data, the no-fraud observations are scattered along the dotted red line. ML classifier modelling on imbalanced data will lead to biased results, with the algorithms only reading the no-fraud observations[21][42]. Oversampling with SMOTe-ENN will create new, synthetic data points similar to the existing minority class.

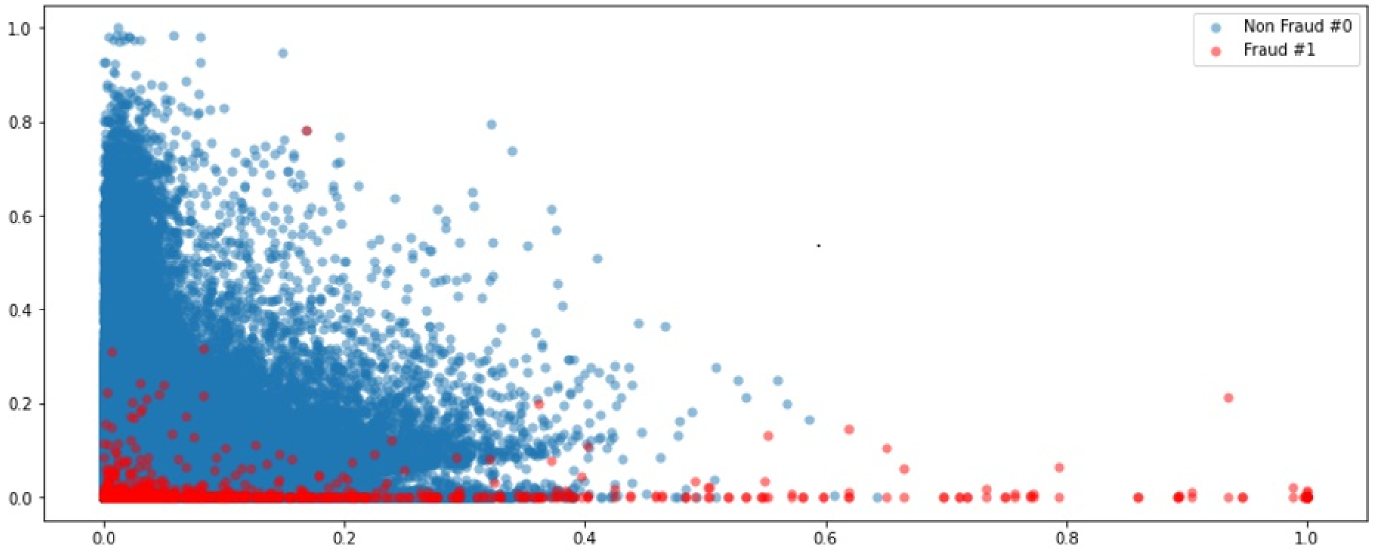


Figure 1. *Data before SMOTe-ENN Application*

Figure 2 shows the data after SMOTe-ENN oversampling. Note that the data on the red line is more densely packed and is now moving in the same direction as the blue line. This symmetry of the lines is caused by the synthetic data points generated by SMOTe-ENN. These data points are not identical to the actual data points but are close enough to be used to train the model[48][49]. The resultant effect is a more accurate model, better able to classify the new data points. By artificially generating additional data points, SMOTe-ENN can ensure that the model can learn from the data and generalize to new inputs (Chawla et al., et al., 2002). In this way, SMOTe-ENN can help to improve the performance of ML models on imbalanced datasets[48][49].

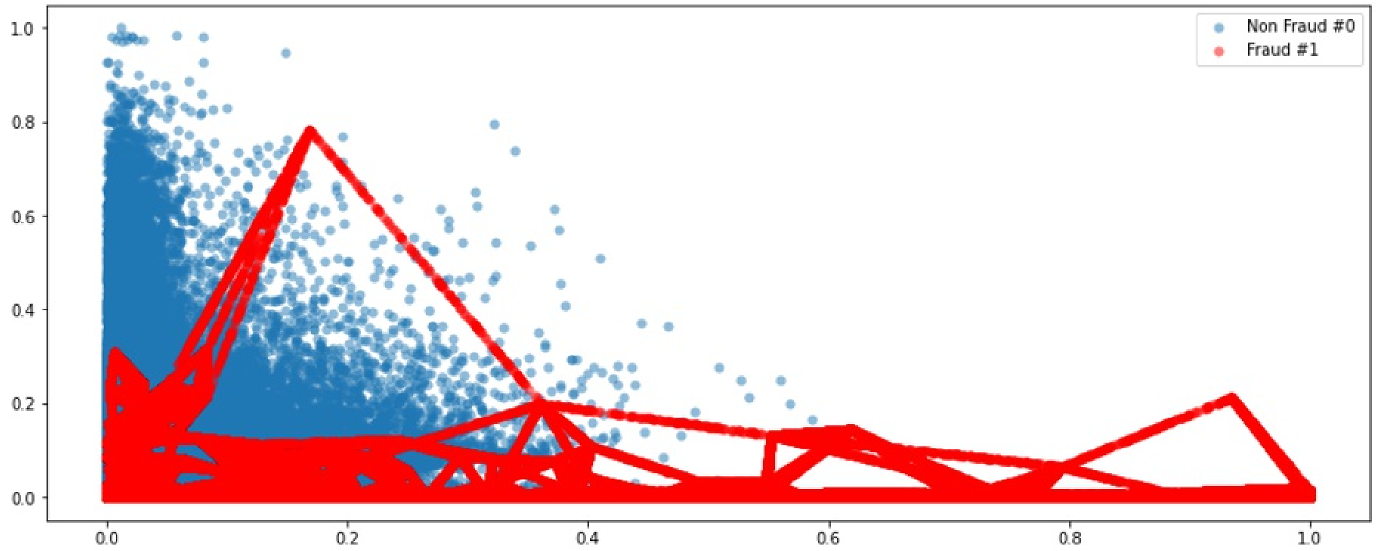


Figure 2. *Data After SMOTE-ENN Application*

3.3.3. Fraud Model Performance Measures

The results were analyzed using the standard confusion matrix. A confusion matrix is a table that helps calculate the accuracy of a classification model and the precision, recall, and f1-score. The table is made up of true positive (TP), false positive (FP), false negative (FN), and true negative (TN) values[17]. For a 2x2 binary classification, the confusion matrix as it pertains to this project can be deciphered as follows:

True Positive (TP): The algorithm predicts the fraud, and the outcome is fraud.

True Negative (TN): The algorithm predicts no fraud and there was no fraud.

False Positive (FP): The algorithm predicted fraud, but there was no fraud (Type 1 Error).

False Negative (FN): The algorithm predicted no-fraud, but there was fraud (Type II Error)

Table 2 presents the performance measures used in the models for this paper. Accuracy is the proportion of correct predictions over all predictions. However, accuracy is not the best metric for an imbalanced dataset. An improved single measure is the Matthews Correlation Coefficient (MCC)[52]. The MCC is a measure of the quality of binary classification.

The MCC considers true and false positives and negatives and is widely regarded as a balanced measure that can be applied even when the classes are of very different sizes[53]. The MCC is in the range [-1, 1]. A coefficient of +1 represents a perfect prediction, a coefficient of 0 represents an average random prediction, and a coefficient of -1 represents an inverse prediction[54]. The MCC has some valuable properties that make it more suitable than other measures for some purposes, most notably its ability to work well even when one of the two classes is much more frequent than the other[52][53]. The formula for both the performance accuracy and the MCC is shown in Table 2.

3.3.4. Practitioner Measures

To understand how well a classification model is performing, we need to look beyond the performance accuracy. The accuracy may be high, but this could be due to the model predicting the most frequent class all the time, which produces inconsistent and unreliable results for an imbalanced dataset. To get a better idea of model performance, the confusion matrix can be used to calculate the performance of other metrics. From the confusion matrix, we can calculate the precision, recall, and F1-score. Precision is the number of correct predictions divided by the total number of predictions. The recall is the number of correct predictions divided by the total number of actual positive cases. Generally, a classifier with a higher precision but lower recall will miss some fraudulent items but will not incorrectly predict too many items as fraud. A classifier with a higher recall but lower precision will correctly identify more of the fraud items and incorrectly predict more items as being a fraud. The ideal classifier would have perfect precision and recall, but this is usually impossible in practice. Instead, the goal is usually to find a balance between precision and recall that gives the best overall results. The F-1 score is the harmonic mean of precision and recall that achieves this objective. A good classification model will have a high precision, recall, and F1 score[21].

A more robust measure is the Receiver Operating Characteristics (ROC). The ROC curve is a graphical tool used to evaluate the performance of a binary classifier. The curve is generated by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold values. The area under the ROC curve (AUROC) is a metric that can be used to compare different classifiers. A classifier with a higher AUC will have better discrimination, meaning it can better distinguish between positive and negative observations. A perfect classifier would have a TPR of 1 and an FPR of 0, resulting in a point in the upper left corner of the ROC curve. Generally, the closer the ROC curve is to this corner, the better the classifier performs. Classifiers that perform similarly to random guessing will have a ROC curve close to the diagonal line[20][21].

Table 2. Performance Metrics and Formulae

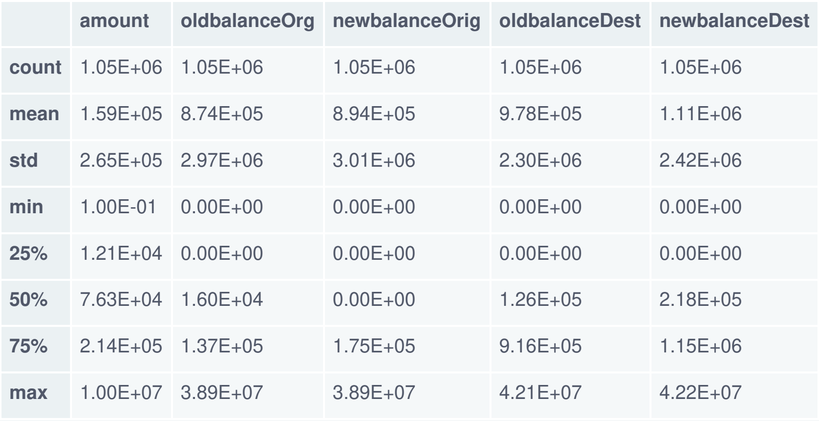
Performance Metrics	Formulae
Accuracy	(TP+TN)/(TP+TN+FP+FN)
MCC	(TP*TN – FP*FN) / √(TP+FP)(TP+FN)(TN+FP)(TN+FN)
Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
F1-score	2*((precision*recall)/(precision + recall))
ROC Curve	Plot TPR (TP/TP+FN) and FPR (FP/TN+FP)

4. Results and Analysis

4.1. Descriptive Results

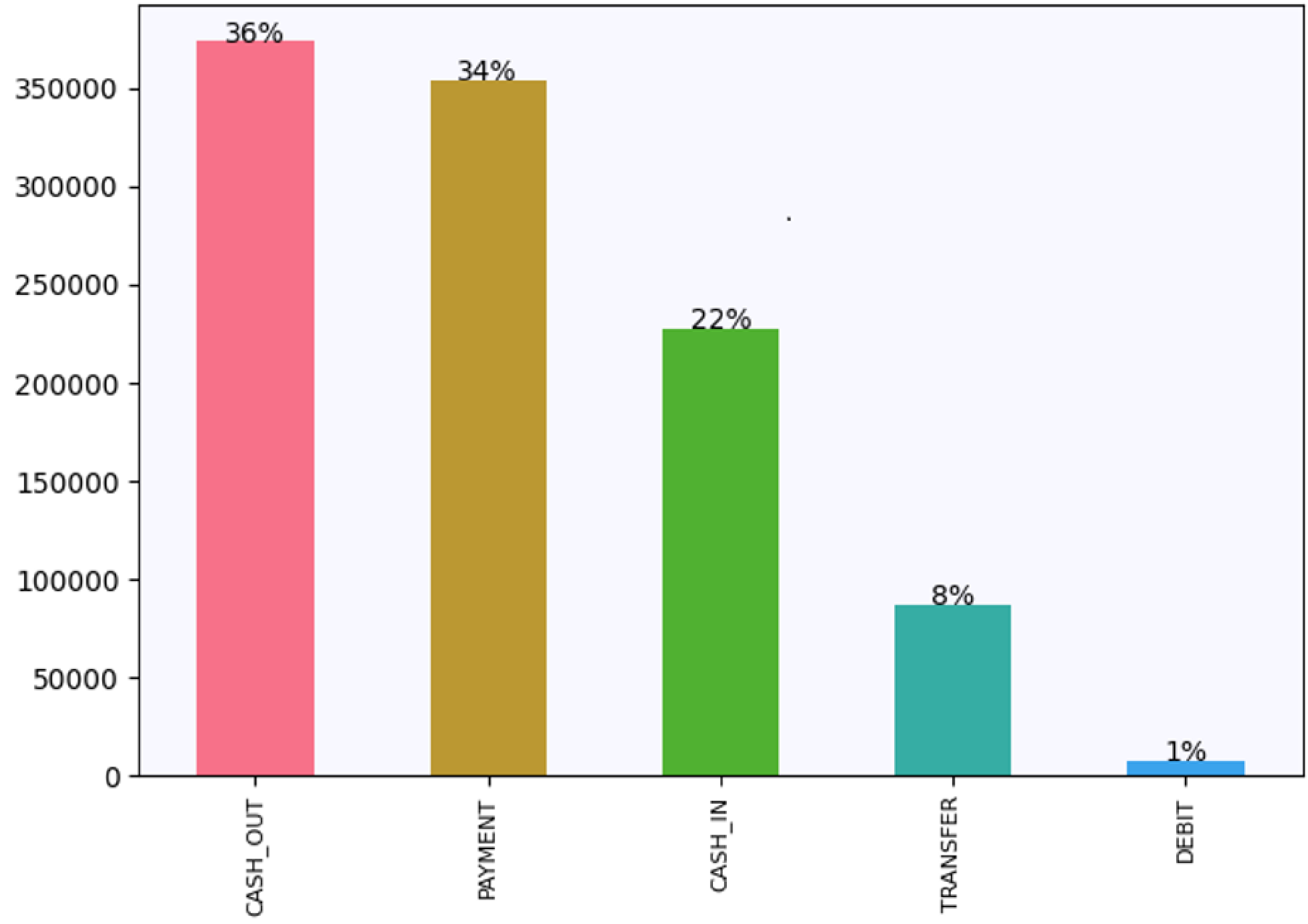
The table provides information on the results of numerical features for money laundering transactions. The average amount of money transferred was $159,000—about $76,000, or 50% of total laundered transactions, made up most of the transfers. The highest amount laundered was $10 million. The average initial balance before the transaction was $875,000, while the average new balance after the transaction was $895,000. Note also that the maximum initial balance before the transfer and the new balance after the transfer was $38 million. The average initial balance of the recipient before the transaction was $978,000, while the average new balance after the transaction was higher at $1.1 million. Once again, it is worth noting that the maximum initial balance and the new balance of the recipient after the transfer was over $40 million. These findings suggest that money laundering is a significant problem, with large sums of money being transferred illegally[14][42].

The findings of this study have implications for both the government and the public. First, the findings indicate that much illegal activity is not being detected or stopped[42]. Second, large amounts of money are not being taxed, which means the government is losing a lot of revenue[55]. Third, the findings prove that an underground economy operates outside the legal system[55][56]. A large underground economy can have several negative consequences, such as making it more difficult for law-abiding businesses to compete and increase the chances of crime and corruption[57]. Regulators need to take action to address the problem of the underground economy or risk people being doubtful of the government's ability to regulate the financial system.



	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest
count	1.05E+06	1.05E+06	1.05E+06	1.05E+06	1.05E+06
mean	1.59E+05	8.74E+05	8.94E+05	9.78E+05	1.11E+06
std	2.65E+05	2.97E+06	3.01E+06	2.30E+06	2.42E+06
min	1.00E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
25%	1.21E+04	0.00E+00	0.00E+00	0.00E+00	0.00E+00
50%	7.63E+04	1.60E+04	0.00E+00	1.26E+05	2.18E+05
75%	2.14E+05	1.37E+05	1.75E+05	9.16E+05	1.15E+06
max	1.00E+07	3.89E+07	3.89E+07	4.21E+07	4.22E+07

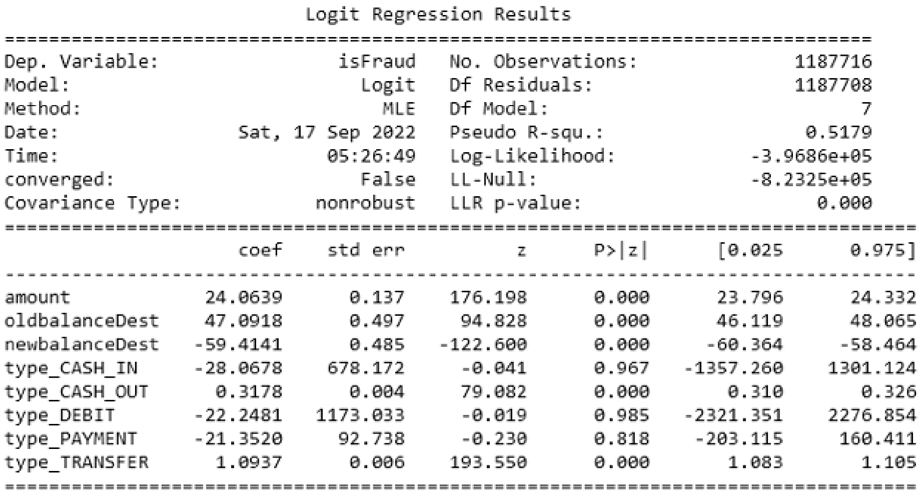
As can be seen in Figure 1, there are five unique types of transactions: cash-out, cash-in, payment, transfer, and debit. Cash-out is the most frequent money laundering transaction, accounting for nearly 36% of all transactions. Cash-out is followed by payment (34%), cash-in (22%), transfer (8%), and debit (1%). As expected, most transactions are in the form of cash (70%), with the remainder made up of payments, transfers, and debit cards. As noted above, the average amount laundered per transaction is $159,000, with a standard deviation of $265,000. These results show that money laundering is usually done with large sums of money, though the amount washed in each transaction varies greatly[6][42].



4.2. Analytical Results

4.2.1. Baseline Logistic Regression

In this study, we sought to compare the performances of several different classifiers to determine which would be best suited for predicting financial transaction fraud. Logistic regression was used as the base model, then compared with the results of other classifiers. As shown in Table 1, all the features except cash-in, debit, and payment type are statistically significant at a *p-value* of 0.05 and with a confidence interval of 95%. These findings suggest that the other classifiers may be more accurate in predicting fraud than the logistic regression model[27]. More research is needed, but the logistic regression results suggest that other classifiers should be considered when predicting financial transaction fraud. To build the different classifiers and to see how the model performs using the classification metrics precision, recall, and the F1-score, the features with *a p-value* > 0.05 will be dropped from the dataset.



```
                           Logit Regression Results
==============================================================================
Dep. Variable:                isFraud   No. Observations:              1187716
Model:                          Logit   Df Residuals:                  1187708
Method:                           MLE   Df Model:                            7
Date:                Sat, 17 Sep 2022   Pseudo R-squ.:                  0.5179
Time:                        05:26:49   Log-Likelihood:            -3.9686e+05
converged:                      False   LL-Null:                   -8.2325e+05
Covariance Type:            nonrobust   LLR p-value:                     0.000
==================================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
amount            24.0639      0.137    176.198      0.000      23.796      24.332
oldbalanceDest    47.0918      0.497     94.828      0.000      46.119      48.065
newbalanceDest   -59.4141      0.485   -122.600      0.000     -60.364     -58.464
type_CASH_IN     -28.0678    678.172     -0.041      0.967   -1357.260    1301.124
type_CASH_OUT      0.3178      0.004     79.082      0.000       0.310       0.326
type_DEBIT       -22.2481   1173.033     -0.019      0.985   -2321.351    2276.854
type_PAYMENT     -21.3520     92.738     -0.230      0.818    -203.115     160.411
type_TRANSFER      1.0937      0.006    193.550      0.000       1.083       1.105
==================================================================================
```

The coefficients of the logistic regression model are in terms of log(odd). The log(odd) in Table 1 lacks interpretation since it does not directly give the odds of an event occurring. Instead, they show how much each feature contributes to the model, meaning that they are more likely to predict whether or not an event will occur. Based on the logistic regression model, the amount involved in the transfer is the most important feature in detecting fraud. The features that positively affect fraud detection are amount, cash-out, and transfer. The features which negatively affect fraud prediction are the initial balance of the recipient before the transaction (i.e., oldbalanceDest) and the new balance of the recipient account after the transaction (i.e., newbalanceDest). However, these coefficients cannot be directly interpreted without first taking the exponential of the features. To find the odds, we must take the exponential of the coefficients. For example, if we take the exponential of-0.693, we get 0.5, which means that for every unit increase in X_1, the odds of $y = 1$ decrease by 0.5. Holding all other variables constant, a one-unit increase in X_1 leads to a 50% decrease in the odds of $y = 1$.

Features	Odds	Change_odd%
amount	7.97E+03	797239.66945
oldbalanceDest	2.01E-07	-99.99998
newbalanceDest	5.63E-03	-99.43677
type_CASH_OUT	1.64E+00	64.21473
type_TRANSFER	4.29E+00	329.07871

Table 1 provides an overview of the odds of the features and how they affect fraud. Note that the initial balance of the recipient before the transaction and transfer type have the highest odds of predicting fraud. Therefore, these two features are important when trying to predict fraud. However, we should also consider other factors, such as the amount of money being transferred, the country where the recipient is located, and whether or not the recipient has a bank account. These features can provide additional insights into whether or not a transaction is fraudulent[42]. For instance, if the amount of money being transferred is very large, it is more likely to be fraudulent. Similarly, this is another red flag if the recipient is located in a country with a high risk of fraud.

Features	odds
oldbalanceDest	8.47E+12
amount	4.20E+08
type_TRANSFER	6.11E+04
type_CASH_OUT	2.65E+04
newbalanceDest	3.42E-19

4.2.2. Performance Accuracy and Matthew Correlation Coefficient

The models were analyzed using the SK learn library after dropping the features with p-values greater than 0.05. Table 1 presents the performance accuracy and the MCC results for the different models tested. Compared to the gradient descent and the ensemble classifier, the benchmark logistic regression model did not fear much. The ensemble classifier, a combination of multiple models, showed the best performance in terms of accuracy and MCC. The superior performance of the ensemble classifiers is because they can capture different patterns in the data and weigh them appropriately, resulting in more accurate predictions[43][58]. In terms of computational cost, gradient descent was the most expensive model to run, while logistic regression was the least expensive[36]. Therefore, if computational cost is a concern, logistic regression may be a better option despite its slightly lower accuracy. Overall, all the models performed reasonably well, with ensembles being the best performers in terms of accuracy and MCC.

As shown in Table 1, the MCC had poorer performance overall than the model accuracy. The random forest model had the highest performance accuracy (.89) and MCC (.78) on the test set, although it showed signs of overfitting. The next best performing classifier was the decision tree classifier, with a performance accuracy of .86 and an MCC of .75. The gradient descent classifier had a similar accuracy (.82); but a lower MCC (.67). These results suggest that while the random forest model is more likely to overfit the data, it may still be the best option for this dataset due to its higher accuracy. However, further tuning of hyperparameters may be necessary to improve its performance. These results suggest that, while gradient descent may be a computationally efficient method, it is not as effective as other methods in predictive power. In conclusion, the random forest model is the best-performing model in terms of accuracy and MCC.

Algorithm	Performance Accuracy		MCC	
	Train	Test	Train	Test
Logistic Regression	0.82	0.83	0.66	0.68
Gradient Descent	0.8	0.82	0.66	0.67
Decision Tree	0.83	0.86	0.71	0.75
Random Forest	1.0	0.89	0.99	0.78

4.2.3. Classification Measures

When dealing with imbalanced datasets, it is important to use measures of performance that are more robust than simple accuracy[42]. The performance accuracy only measures the FP and FN and does not provide enough information to properly assess the model's performance. When dealing with imbalanced datasets, it is important to use measures of performance that are more robust than simple accuracy[39]. In addition, we need to consider the TN and TP rates because these measures provide a complete picture of the model's performance. For example, if we have a high TN rate, our model correctly identifies negative instances. On the other hand, if we have a low TN rate, our model incorrectly identifies positive instances. As a result, the TN and TP rates are significant for assessing the performance of imbalanced datasets[17].

When choosing a machine learning model for fraud detection, it is important to consider how well the model will perform in terms of precision, recall, and the F1- score[42][51]. These measures are better indicators of how well a model predicts laundered transactions than accuracy alone. However, it is important to remember that all these measures can be affected by the choice of threshold. A high threshold will result in fewer FPs and TPs, while a low threshold will have the opposite effect. Therefore, tuning the threshold according to the application's needs is important. In some cases, it may even be necessary to use multiple thresholds to achieve the desired level of performance[22][44]. To offset the trade-off between precision and recall, the threshold was tuned to ensure that it was equal before running the classifiers.

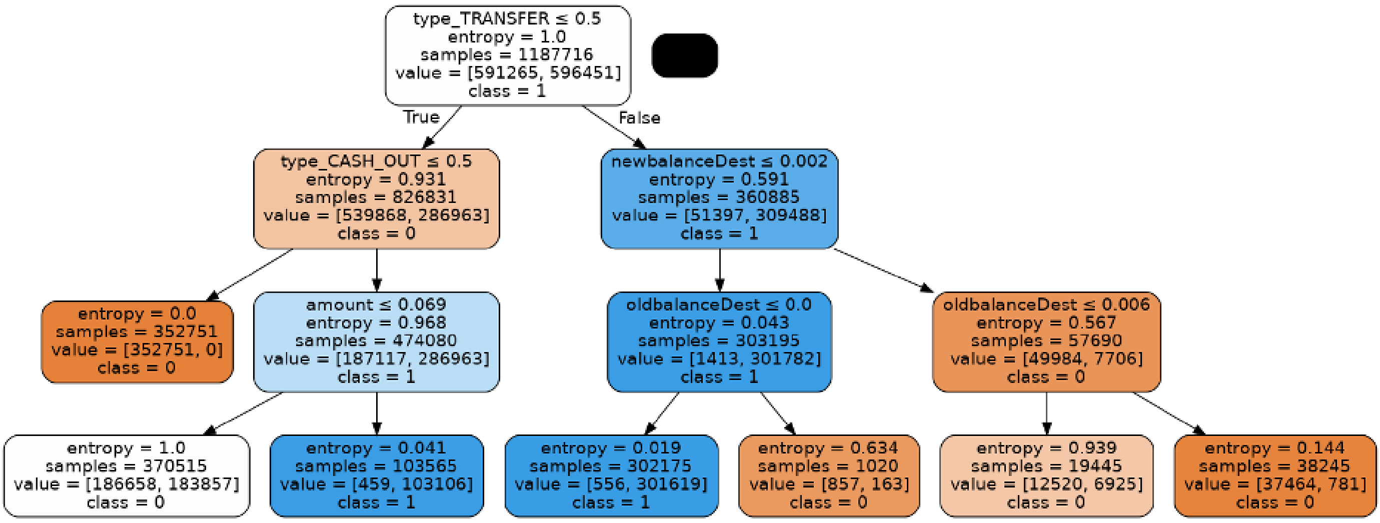
Table 1 shows the results of the classifiers' performance. Note that the decision tree classifier had the highest precision score (1.0), followed by the random forest model (.96). In contrast, the logistic regression and gradient classifiers had the highest recall scores (.96). The random forest (.87) followed by logistic regression (.85) had the highest F1 scores. The gradient descent and decision tree classifiers had the lowest F1- scores, respectively (.84). These results indicate that the random forest is the best performing classifier overall because the model achieved high scores across all three measures. Contextually, these results can be interpreted to mean that the random forest classifier correctly identifies a high proportion of positives and negatives and achieve a high degree of overall accuracy. The logistic regression and gradient descent classifiers also performed well, achieving high scores in recall and F1. However, they did not achieve the same high precision score as the random forest classifier. These results suggest that the logistic regression and gradient descent classifiers may misclassify more cases than the random forest classifier.

Classification Metrics

Algorithm	Precision	Recall	F1-Score
Logistic Regression	0.76	0.96	0.85
Gradient Descent	0.75	0.96	0.84
Decision Tree	1.0	0.72	0.84
Random Forest	0.96	0.80	0.87

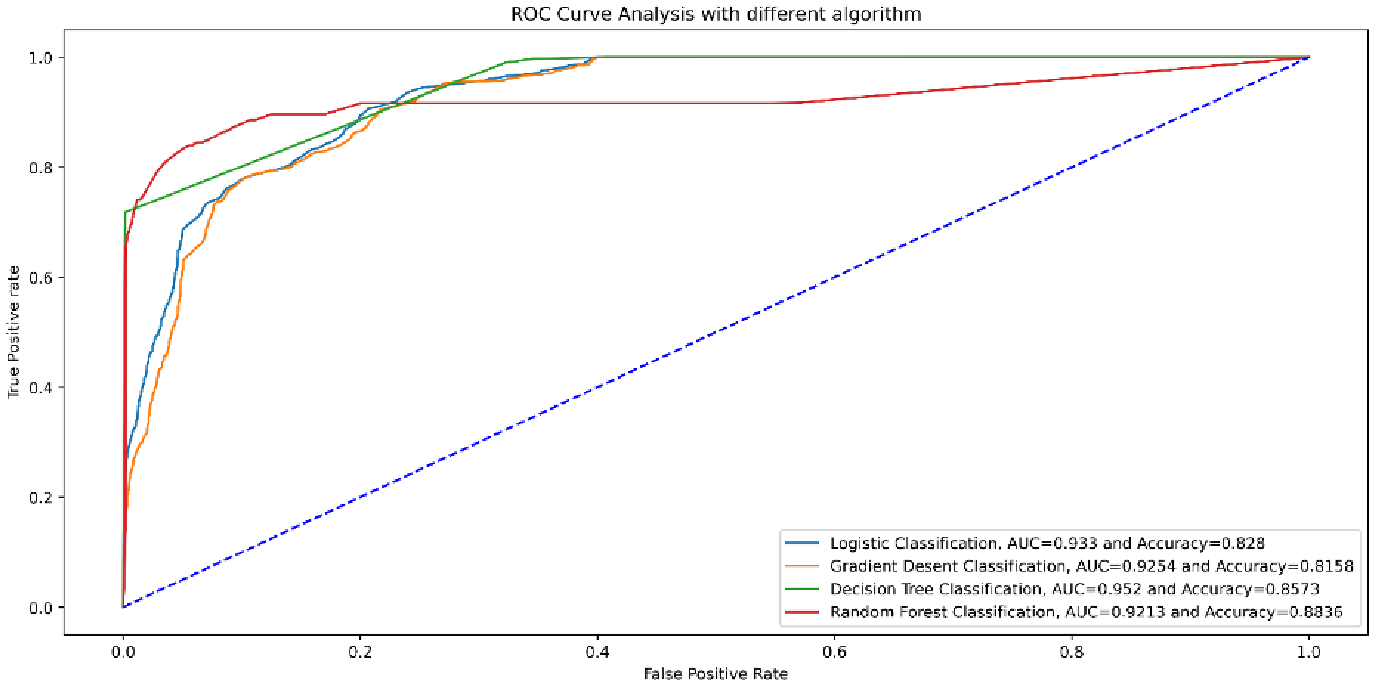
As shown in Table 1, the decision tree classifier had the highest precision score, which means that the decision tree model did an excellent job predicting the fraudulent transactions. As shown in Figure 1, the type of transfer was the feature selected to split the tree. If the amount laundered is ≤0.5, the tree splits at the amount that was cashed out, and if the amount is ≥ 0.5, the tree splits at the amount the recipient had after the laundered transaction (newbalanceDest). The 'type of transfer' is an important feature in classifying a money laundering transaction because it can show how sophisticated or structured the laundering process is. For example, a cash-out is a transfer with an influx of money into one account and then an immediate withdrawal of those funds. This type of laundering is generally associated with low-level or first-time offenders.

In contrast, a structured deposit is when funds are gradually deposited into an account over time before being withdrawn. This type of laundering is generally associated with more knowledgeable or experienced offenders with access to multiple accounts. The decision tree correctly classified 84% of all cash-outs as money laundering transactions and 100% of all structured deposits as money laundering transactions. These results indicate that the decision tree is an effective classifier for identifying money laundering activities. Both the accuracy scores for cash-outs and structured deposits are high, suggesting that the decision tree could be improved by adding more information, such as the customer's location or account history.



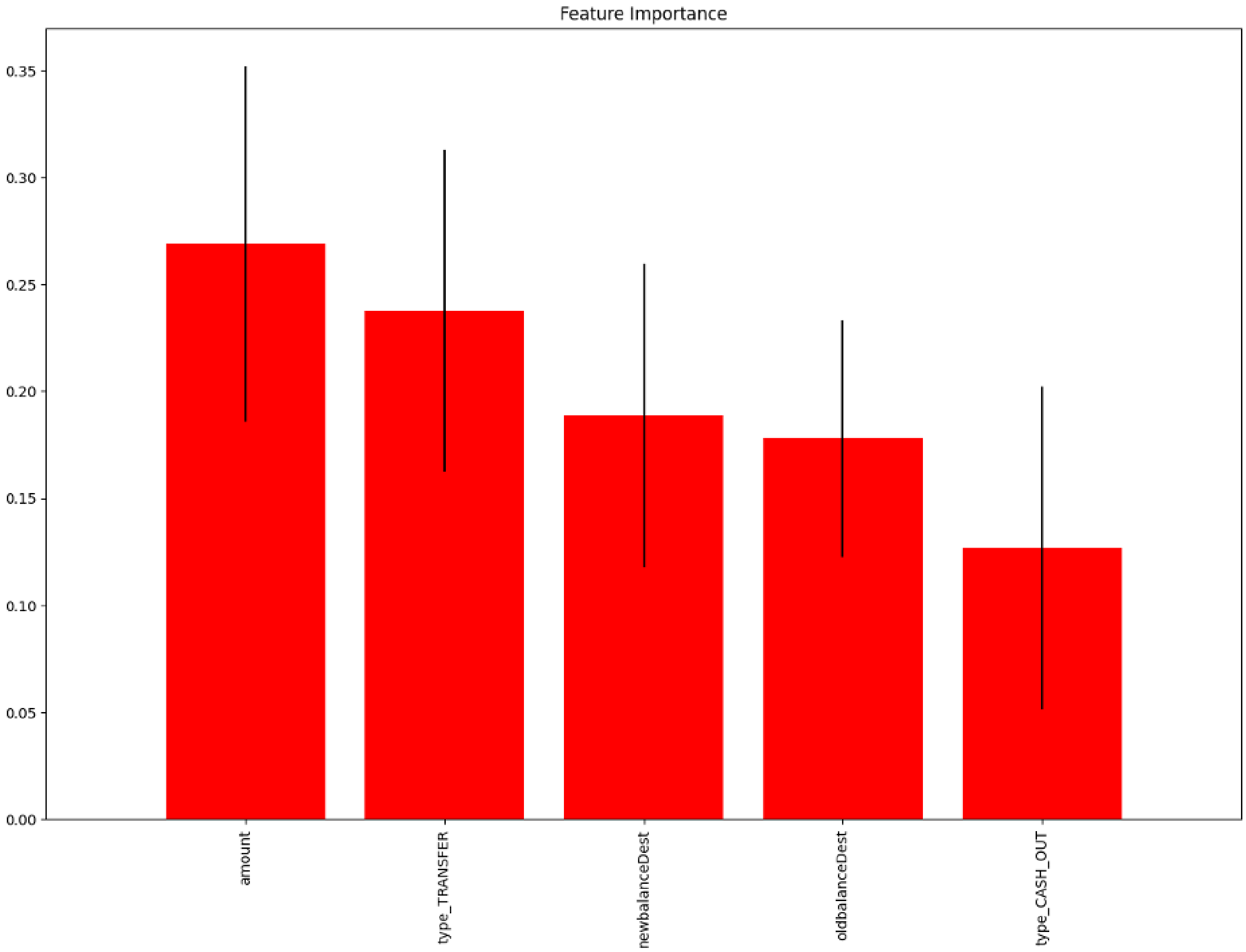
4.2.4. The ROC Curve

The ROC curve is a commonly used metric for evaluating the performance of classification models, mainly when dealing with imbalanced datasets[51]. As seen in Figure 1, the AUROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold values and provides a visual representation of the model's accuracy. The decision tree model was the best performing classifier (.95), followed by the logistic regression model (.93). The random forest and gradient descent classifiers had the lowest AUC score of .92. These findings suggest that, while all four models are effective at predicting laundered transactions, the decision tree model is the most accurate. Furthermore, the logistic regression model is a close second, making it a good choice for businesses that require a more interpretable model. Finally, the random forest and gradient descent models are still effective predictors of fraud, but they are not as accurate as the other two models.



4.3. Feature Importance

The features with the highest p-values in the baseline logistic regression model were amount, oldbalanceDest, newbalanceDest, cash-out, and transfer. These features were compared with the features from the ensemble classifiers to examine which ones contributed more to predicting laundered transactions. As shown in Figure 1, the amount involved in the transfer was again the top feature to predict suspicious transactions in mobile money transfers. These findings suggest that the amount of a transaction is a good predictor of whether a transaction is suspicious or not. The other features with p-values > 0.05 did not contribute much to predicting suspicious transactions. Altogether, mobile money transfer providers need to be aware of transfers involving large amounts of money and those made frequently or without a specific destination[32]. By considering these factors, mobile money providers can detect and prevent suspicious behaviour on their platforms.



5. Discussion and Conclusion

The popularity of mobile money transfer services has grown rapidly in recent years, driven by the increasing penetration of mobile devices and the widespread adoption of mobile banking services. However, using mobile money transfer services has also created new opportunities for criminals to launder money. To provide more insights into this problem, this study employs ML classifiers to predict mobile money transfer laundering transactions[34][42]. While all the classifiers were very useful in predicting suspicion transactions, the random forest model was the most consistent and best-performing model across all the classifiers[43]. The random forest model is a powerful ML technique for fraud detection because it can effectively learn from data with many features and can be tuned to achieve high precision (.96), recall (.80), and F1-score (.87). The findings from this study provide useful intelligence for mobile money service providers and law enforcement agencies to fight against money laundering.

Given the current global landscape, it is not surprising that the use of mobile money has increased dramatically in the past few years[55]. For consumers and businesses alike, the convenience and flexibility of mobile wallets have made them a popular choice[18]. However, the rise of mobile money has also attracted the attention of criminals looking for new ways to launder their ill-gotten gains[1][20][34]. In many cases, mobile money services are being used to facilitate money laundering by allowing criminals to quickly and easily move large amounts of cash without raising suspicion. For example, recent studies found that a significant proportion of mobile money users have been involved in a transaction that could be considered suspicious[8][16][46][47]. Considering these findings, more must be done to prevent mobile money from being used for illegal purposes. While mobile money has brought many benefits to the global economy, it is important to remember that those with criminal intent can also exploit its services.

Financial institutions have constantly been pressured to prevent money laundering and terrorist financing by implementing stringent compliance measures[13][55]. The mobile money ecosystem has only added to this pressure, as financial institutions must now also contend with the challenges posed by digital transactions. In response, there has been a shift in emphasis from traditional, deterministic rules-based methodologies toward more sophisticated computational techniques. This shift is primarily because the sheer volume of transaction data makes it difficult to flag and detect suspicious activities using rules-based approaches. Computational techniques in the form of ML offer a more effective way to monitor suspicious behaviour, as they can consider a broader range of factors and larger datasets. Adopting ML techniques for money laundering detection is a promising development in the fight against illicit activities. These techniques could make catching people trying to launder money through criminal networks much more accessible. They could also help mobile money service providers and law enforcement agencies stay one step ahead of individuals trying to launder illicit funds.

5.1. Limitations and Future Research

While ML is a valuable tool for fraud detection, several limitations must be considered when conducting research in this area. First, the data sets used to train the algorithms may not represent the population of interest, leading to inaccurate predictions. Second, ML models can be biased based on previous studies that used flawed methodology. Third, it is important to remember that ML is only one tool for detecting fraud; other methods, such as human intelligence and expert analysis, may be more effective in certain situations. Despite these limitations, ML is a powerful tool that significantly impacts fraud research. With continued advances in this area, even more, progress will likely be made in using ML algorithms to fight against fraud. Future research can employ ML to identify behaviour patterns that may indicate fraudulent activity, while human intelligence can provide context and insights that an automated system may miss. By combining these two approaches, future research can significantly improve the fight against fraud.

References

1. [a, b, c] *Aron J. Mobile Money and the Economy: A Review of the Evidence. The World Bank Research Observer. 2018 Aug 1;33(2):135–88.*
2. [^] *Reaves B, Bowers J, Scaife N, Bates A, Bhartiya A, Traynor P, et al. Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications. ACM Trans Priv Secur. 2017 Aug 11;20(3):1–31.*
3. [a, b] *Zhdanova M, Repp J, Rieke R, Gaber C, Hemery B. No Smurfs: Revealing Fraud Chains in Mobile Money Transfers. In: 2014 Ninth International Conference on Availability, Reliability and Security [Internet]. Fribourg, Switzerland: IEEE; 2014 [cited 2022 Sep 21]. p. 11–20. Available from: http://ieeexplore.ieee.org/document/6980259/*
4. [^] *Rieke R, Zhdanova M, Repp J, Giot R, Gaber C. Fraud Detection in Mobile Payments Utilizing Process Behavior Analysis. In: 2013 International Conference on Availability, Reliability and Security [Internet]. Regensburg, Germany: IEEE; 2013 [cited 2022 Sep 21]. p. 662–9. Available from: https://ieeexplore.ieee.org/document/6657303/*
5. [^] *Nyamtiga BW, Sam A, Laizer LS. Enhanced Security Model For Mobile Banking Systems In Tanzania. 2013;1(4):17.*
6. [a, b, c, d] *Merritt C. Mobile money transfer services: The next phase in the evolution of person-to-person payments. Journal of Payments Strategy & Systems. 2011 Jun 1;5(2):143–60.*
7. [^] *Lake AJ. Risk Management in Mobile Money: Observed Risks and Proposed Mitigants for Mobile Money Operators [Internet]. World Bank; 2013 [cited 2022 Sep 21]. Available from: http://elibrary.worldbank.org/doi/book/10.1596/28420*
8. [a, b, c] *Ali G, Ally Dida M, Elikana Sam A. Evaluation of Key Security Issues Associated with Mobile Money Systems in Uganda. Information. 2020 Jun 8;11(6):309.*
9. [^] *Kikulwe EM, Fischer E, Qaim M. Mobile Money, Smallholder Farmers, and Household Welfare in Kenya. PLOS ONE. 2014 Oct 6;9(10):e109804.*
10. [^] *Carmi G, Segal SY. Mobile Security: a Review of New Advanced Technologies to Detect and Prevent E-Payment Mobile Frauds. Mobile Security. 2016, 3(4), 292-302.*
11. [^] *Kanobe F, Alexander PM, Bwalya KJ. Policies, Regulations and Procedures and their Effects on Mobile Money Systems in Uganda. THE ELECTRONIC JOURNAL OF INFORMATION SYSTEMS IN DEVELOPING COUNTRIES. 2017;83(1):1–15.*
12. [^] *Novikova E, Kotenko I. Visual Analytics for Detecting Anomalous Activity in Mobile Money Transfer Services. In: Teufel S, Min TA, You I, Weippl E, editors. Availability, Reliability, and Security in Information Systems. Cham: Springer International Publishing; 2014. p. 63–78. (Lecture Notes in Computer Science).*
13. [a, b, c] *Solin M, Zerzan A. Mobile Money: Methodology for Assessing Money Laundering and Terrorist Financing Risks. 2010, https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2012/03/amlfinal35.pdf*
14. [a, b] *Akomea-Frimpong I, Andoh C, Akomea-Frimpong A, Dwomoh-Okudzeto Y. Control of fraud on mobile money services in Ghana: an exploratory study. JMLC. 2019 May 7;22(2):300–17.*
15. [^] *Gaber C, Hemery B, Achemlal M, Pasquet M, Urien P. Synthetic logs generator for fraud detection in mobile transfer services. In: 2013 International Conference on Collaboration Technologies and Systems (CTS). 2013. p. 174–9.*
16. [a, b] *Mtaho A. Improving Mobile Money Security with Two-Factor Authentication. IJCA. 2015 Jan 16;109(7):9–15.*
17. [a, b, c] *Singh A, Jain A, Biable SE. Financial Fraud Detection Approach Based on Firefly Optimization Algorithm and Support Vector Machine. Ramachandran M, editor. Applied Computational Intelligence and Soft Computing. 2022 Jun*

17;2022:1–10.

18. a, b, c, d, e *Kang H. Fraud Detection in Mobile Money Transactions Using Machine Learning. Information Systems and Business Analytics. 2019;5(32):320–32.*
19. a, b *Singh K, Best P. Anti-Money Laundering: Using data visualization to identify suspicious activity. International Journal of Accounting Information Systems. 2019 Sep 1;34:100418.*
20. a, b, c, d, e, f, g, h, i *Botchey FE, Qin Z, Hughes-Lartey K. Mobile Money Fraud Prediction—A Cross-Case Analysis on the Efficiency of Support Vector Machines, Gradient Boosted Decision Trees, and Naïve Bayes Algorithms. Information. 2020 Aug;11(8):383.*
21. a, b, c, d, e, f, g, h *Lokanan ME, Sharma K. Fraud prediction using machine learning: The case of investment advisors in Canada. Machine Learning with Applications. 2022 Jun 15;8:100269.*
22. a, b, c, d, e, f, g *Lokanan M, Liu S. Predicting Fraud Victimization Using Classical Machine Learning. Entropy. 2021 Mar;23(3):300.*
23. ^ *Pech R. Fraud detection in mobile money transfer as binary classification problem. 2019, https://www.researchgate.net/profile/Ratha-Pech/publication/333755188_Fraud_detection_in_mobile_money_transfer_as_binary_classification_problem/links/5d0251554585157d15a71229/Fraud-detection-in-mobile-money-transfer-as-binary-classification-problem.pdf*
24. a, b *Aslam N, Khan IU, Alansari A, Alrammah M, Alghwairy A, Alqahtani R, et al. Anomaly Detection Using Explainable Random Forest for the Prediction of Undesirable Events in Oil Wells. Ramachandran M, editor. Applied Computational Intelligence and Soft Computing. 2022 Aug 5;2022:1–14.*
25. a, b, c, d, e, f, g *Bagga S, Goyal A, Gupta N, Goyal A. Credit Card Fraud Detection using Pipeline and Ensemble Learning. Procedia Computer Science. 2020;173:104–12.*
26. a, b, c *Dighe D, Patil S, Kokate S. Detection of Credit Card Fraud Transactions Using Machine Learning Algorithms and Neural Networks: A Comparative Study. In: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). 2018. p. 1–6.*
27. a, b, c, d, e *Itoo F, Meenakshi, Singh S. Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. Int j inf tecnol. 2021 Aug 1;13(4):1503–11.*
28. ^ *Perols J. Financial Statement Fraud Detection: An Analysis of Statistical and Machine Learning Algorithms. AUDITING: A Journal of Practice & Theory. 2011 May 1;30(2):19–50.*
29. a, b *Sundarkumar GG, Ravi V, Siddeshwar V. One-class support vector machine based undersampling: Application to churn prediction and insurance fraud detection. In: 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC). 2015. p. 1–7.*
30. ^ *Herland M, Khoshgoftaar TM, Bauder RA. Big Data fraud detection using multiple medicare data sources. J Big Data. 2018 Dec;5(1):29.*
31. a, b *Thornton D, Mueller RM, Schoutsen P, van Hillegersberg J. Predicting Healthcare Fraud in Medicaid: A Multidimensional Data Model and Analysis Techniques for Fraud Detection. Procedia Technology. 2013 Jan 1;9:1252–64.*
32. a, b *Coppolino L, D'Antonio S, Formicola V, Massei C, Romano L. Use of the Dempster-Shafer Theory for Fraud*

Detection: The Mobile Money Transfer Case Study. In: Camacho D, Braubach L, Venticinque S, Badica C, editors. Intelligent Distributed Computing VIII. Cham: Springer International Publishing; 2015. p. 465–74. (Studies in Computational Intelligence).

33. ^*Sahin Y, Duman E. Detecting credit card fraud by ANN and logistic regression. In: 2011 International Symposium on Innovations in Intelligent Systems and Applications. 2011. p. 315–9.*
34. a, b, c*Bashir S, Ghous DH. Detecting Mobile Money Laundering Using Genetic Algorithm as Feature Selection Method with Classification Method. LC International Journal of STEM (ISSN: 2708-7123). 2020;1(4):121 129-121 129.*
35. a, b*Sahin Y, Bulkan S, Duman E. A cost-sensitive decision tree approach for fraud detection. Expert Systems with Applications. 2013 Nov 1;40(15):5916–23.*
36. a, b, c, d*Ruder S. An overview of gradient descent optimization algorithms [Internet]. arXiv; 2017 [cited 2022 Sep 21]. Available from: http://arxiv.org/abs/1609.04747*
37. ^*Mercier Q, Poirion F, Désidéri JA. A stochastic multiple gradient descent algorithm. European Journal of Operational Research. 2018 Dec 16;271(3):808–17.*
38. a, b*Jing R, Tian H, Li Y, Zhang X, Zheng X, Zhang Z, et al. Improving the Data Quality for Credit Card Fraud Detection. In: 2020 IEEE International Conference on Intelligence and Security Informatics (ISI). 2020. p. 1–6.*
39. a, b, c*Duan L. Performance Evaluation and Practical Use of Supervised Data Mining Algorithms for Credit Card Approval. In: 2020 International Conference on Computing and Data Science (CDS). 2020. p. 251–4.*
40. ^*Li Z, Liu G, Jiang C. Deep Representation Learning With Full Center Loss for Credit Card Fraud Detection. IEEE Transactions on Computational Social Systems. 2020 Apr;7(2):569–79.*
41. a, b, c*Jurgovsky J, Granitzer M, Ziegler K, Calabretto S, Portier PE, He-Guelton L, et al. Sequence classification for credit-card fraud detection. Expert Systems with Applications. 2018 Jun 15;100:234–45.*
42. a, b, c, d, e, f, g, h, i, j*Lokanan ME. Predicting Money Laundering Using Machine Learning and Artificial Neural Networks Algorithms in Banks. Journal of Applied Security Research. 2022 Aug 26;0(0):1–25.*
43. a, b, c, d, e, f*Nami S, Shajari M. Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors. Expert Systems with Applications. 2018 Nov 15;110:381–92.*
44. a, b, c*Dornadula VN, Geetha S. Credit Card Fraud Detection using Machine Learning Algorithms. Procedia Computer Science. 2019 Jan 1;165:631–41.*
45. a, b, c, d, e*Lopez-Rojas EA. Applying Simulation to the Problem of Detecting Financial Fraud. 2016 [cited 2022 Sep 21]; Available from: http://urn.kb.se/resolve?urn=urn:nbn:se:bth-12932*
46. a, b, c, d*Rojas EAL, Axelsson S, Baca D. Analysis of fraud controls using the PaySim financial simulator. IJSPM. 2018;13(4):377.*
47. a, b*Lopez-Rojas EA, Barneaud C. Advantages of the PaySim Simulator for Improving Financial Fraud Controls. In: Arai K, Bhatia R, Kapoor S, editors. Intelligent Computing. Cham: Springer International Publishing; 2019. p. 727–36. (Advances in Intelligent Systems and Computing).*
48. a, b, c, d*Luengo J, Fernández A, García S, Herrera F. Addressing data complexity for imbalanced data sets: analysis of SMOTE-based oversampling and evolutionary undersampling. Soft Comput. 2011 Oct 1;15(10):1909–36.*
49. a, b, c, d, e*Aswathi M, Ghosh A, Namboothiri LV. Borda Count Versus Majority Voting for Credit Card Fraud Detection.*

In: Karuppusamy P, Perikos I, García Márquez FP, editors. Ubiquitous Intelligent Systems. Singapore: Springer; 2022. p. 319–30. (Smart Innovation, Systems and Technologies).

50. ^*Almhaithawi D, Jafar A, Aljnidi M. Example-dependent cost-sensitive credit cards fraud detection using SMOTE and Bayes minimum risk. SN Appl Sci. 2020 Aug 27;2(9):1574.*
51. a, b, c, d, e*Sisodia DS, Reddy NK, Bhandari S. Performance evaluation of class balancing techniques for credit card fraud detection. In: 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI). 2017. p. 2747–52.*
52. a, b*Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA) - Protein Structure. 1975 Oct 20;405(2):442–51.*
53. a, b*Chicco D, Jurman G. An Invitation to Greater Use of Matthews Correlation Coefficient in Robotics and Artificial Intelligence. Front Robot AI. 2022 Mar 25;9:876814.*
54. ^*Ryman-Tubb NF, Krause P, Garn W. How Artificial Intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark. Engineering Applications of Artificial Intelligence. 2018 Nov 1;76:130–57.*
55. a, b, c, d*Hendriyetty N, Grewal BS. Macroeconomics of money laundering: effects and measurements. Journal of Financial Crime. 2017 Jan 1;24(1):65–81.*
56. ^*Amoh JK, Adafula B. An estimation of the underground economy and tax evasion: Empirical analysis from an emerging economy. Journal of Money Laundering Control. 2019 Jan 1;22(4):626–45.*
57. ^*Bashlakova V, Bashlakov H. The study of the shadow economy in modern conditions: Theory, methodology, practice. The Quarterly Review of Economics and Finance. 2021 Aug 1;81:468–80.*
58. ^*Zareapoor M, Shamsolmoali P. Application of credit card fraud detection: Based on bagging ensemble classifier. Procedia computer science. 2015;48(2015):679–85.*