Full Financial (Credit and Debit Card) Fraud Detection

Using Machine Learning Classifiers: Decision Tree, K-Nearest Neighbours, Logistic Regression, and Random Forest

Ayomide Ayeni | Cyber Fraud Investigator & Analyst | February, 2025

Abstract

Financial fraud involving credit and debit card transactions represents one of the most pervasive and costly challenges faced by financial institutions globally. This paper presents a comprehensive study on the detection of fraudulent financial transactions using four supervised machine learning classification algorithms: Decision Tree (DT), K-Nearest Neighbours (KNN), Logistic Regression (LR), and Random Forest (RF). A publicly available financial transactions dataset containing over one million records was employed, featuring transaction types including PAYMENT, TRANSFER, CASH_OUT, DEBIT, and CASH_IN. The dataset exhibits extreme class imbalance, with fraudulent transactions constituting only approximately 0.0536% of all records. Following rigorous preprocessing—including feature engineering, categorical encoding, dimensionality reduction, feature scaling via StandardScaler, and a 75:25 train-test split—each model was trained and evaluated using accuracy, precision, recall, F1-score, ROC-AUC, Precision-Recall AUC (AUC-PRC), confusion matrices, and calibration curves. Results demonstrate that the Decision Tree and Random Forest classifiers achieve the highest overall accuracy (99.94% and 99.95% respectively) and both attain an ROC-AUC of 0.84, outperforming KNN (AUC = 0.72) and Logistic Regression (AUC = 0.61). The Random Forest model demonstrated the most balanced precision-recall trade-off for the minority (fraud) class, making it the preferred candidate for real-world deployment.

Index Terms — *Financial Fraud Detection, Credit Card Fraud, Machine Learning, Decision Tree, Random Forest, K-Nearest Neighbours, Logistic Regression, Class Imbalance, ROC-AUC, Feature Engineering.*

I. Introduction

The global financial system is critically dependent on the security and integrity of electronic payment infrastructure. Credit and debit card fraud has grown at an alarming rate with the expansion of digital payment channels, e-commerce platforms, and mobile banking. According to industry reports, global card fraud losses exceeded $33 billion annually, placing significant burdens on financial institutions, merchants, and consumers alike.

Financial fraud detection is inherently a binary classification problem: given a set of transaction attributes, the goal is to determine whether a transaction is fraudulent (class = 1) or legitimate (class = 0). However, this problem is complicated by the extreme rarity of fraudulent events—typically less than 0.1% of all transactions—creating a severe class imbalance that renders naive classifiers ineffective. A classifier that labels every transaction as legitimate can achieve over 99.9% accuracy while detecting zero fraud cases, rendering conventional accuracy metrics misleading.

Machine learning (ML) offers powerful tools for addressing this challenge by learning non-linear decision boundaries and complex feature interactions from historical data. In this work, we systematically implement and compare four widely-used supervised learning algorithms across identical preprocessing pipelines and evaluation protocols, enabling rigorous and fair benchmarking.

The primary contributions of this paper are:

- A complete end-to-end ML pipeline for financial fraud detection, from raw data ingestion through model evaluation.
- Comparative analysis of four distinct classification paradigms on a large-scale, highly imbalanced real-world dataset.
- Multi-dimensional evaluation using accuracy, F1-score, ROC-AUC, AUC-PRC, confusion matrices, and calibration curves.
- Identification of the optimal model configuration for deployment in production fraud detection systems.

II. Related Work

The detection of financial fraud using machine learning has been an active area of research for over two decades. Dal Pozzolo et al. (2015) highlighted the challenges of concept drift and class imbalance in credit card fraud detection, proposing adaptive resampling strategies. Bhattacharyya et al. (2011) demonstrated that support vector machines (SVMs) and random forests outperform logistic regression for fraud detection on skewed datasets.

Deep learning approaches have also gained traction. Fiore et al. (2019) applied generative adversarial networks (GANs) to synthesise minority class samples for data augmentation, while Kim and Kim (2021) explored long short-term memory (LSTM) networks for sequential transaction modelling. However, tree-based ensemble methods such as Random Forest and Gradient Boosting consistently demonstrate competitive performance with significantly lower computational cost.

More recently, graph-based approaches that model relationships between accounts and transactions have shown promise for detecting organised fraud rings (Liu et al., 2022). Despite these advances, classical ML approaches remain the industry standard for production deployment due to their interpretability, lower latency, and robustness. This study contributes to the literature by providing a direct comparative analysis on a large-scale synthetic financial dataset with transparent, reproducible methodology.

III. Dataset Description

The dataset used in this study is a large-scale synthetic financial transactions dataset simulating mobile money transactions. The dataset contains 6,362,620 rows and 11 features prior to preprocessing, spanning 30 simulated days. The following table summarises the dataset attributes.

Feature	Type	Description
step	Integer	Unit of time (1 step = 1 hour); 744 steps total (30 days)
type	Categorical	Transaction type: PAYMENT, TRANSFER, CASH_OUT, DEBIT, CASH_IN
amount	Float	Transaction amount in local currency
nameOrig	String	Originating customer account identifier
oldbalanceOrg	Float	Sender account balance before transaction
newbalanceOrig	Float	Sender account balance after transaction
nameDest	String	Destination customer/merchant account identifier
oldbalanceDest	Float	Recipient balance before transaction
newbalanceDest	Float	Recipient balance after transaction
isFraud	Binary	Target variable: 1 = Fraudulent, 0 = Legitimate

isFlaggedFraud	Binary	System-flagged suspicious transaction (dropped in preprocessing)

Table I: *Dataset Feature Descriptions*

The target variable isFraud reveals a severe class imbalance: 99.9% of transactions are legitimate and only 0.0536% are fraudulent. The visualisation generated during exploratory data analysis confirms this distribution, as shown in the Fraud vs Non-Fraud pie chart where the fraudulent slice is barely visible at 0.0536%. This imbalance necessitates careful selection of evaluation metrics beyond simple accuracy.

IV. Data Preprocessing

A systematic preprocessing pipeline was implemented to prepare the raw dataset for model training. The pipeline consists of the following stages:

A. Feature Selection and Dimensionality Reduction

Six features were removed from the dataset to reduce dimensionality and eliminate features that would cause data leakage or add no predictive value:

```
new_df = df.drop(['step', 'nameOrig', 'nameDest', 'isFlaggedFraud', 'oldbalanceDest',
'newbalanceDest'], axis=1)
```

The retained features are: type, amount, oldbalanceOrg, newbalanceOrig, and isFraud (target). The step feature was excluded as it represents temporal sequencing and may cause overfitting. Account identifiers (nameOrig, nameDest) were removed as high-cardinality nominal variables. isFlaggedFraud was removed to prevent label leakage.

B. Categorical Encoding

The transaction type field is categorical with five unique values. These were mapped to ordinal integers to enable numerical model processing:

```
new_df['type'] = new_df['type'].map({'PAYMENT':1, 'TRANSFER':4, 'CASH_OUT':2, 'DEBIT':5,
'CASH_IN':3})
```

C. Feature and Target Extraction

Features (X) and the target label (y) were extracted from the processed DataFrame:

```
X = new_df.iloc[:,:-1].values y = new_df.iloc[:,-1].values
```

D. Train-Test Split

The dataset was partitioned into training and test sets using a 75:25 ratio with a fixed random seed for reproducibility:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
```

E. Feature Scaling

StandardScaler was applied to standardise features to zero mean and unit variance. The scaler was fitted exclusively on the training set and subsequently applied to the test set to prevent data leakage from the test partition:

```
sc = StandardScaler() X_train = sc.fit_transform(X_train) X_test = sc.transform(X_test)
```

Feature scaling is particularly important for distance-based algorithms such as KNN and gradient-sensitive algorithms such as Logistic Regression. Tree-based models (DT, RF) are scale-invariant but were trained on the scaled data for consistency in this comparative study.

V. Exploratory Data Analysis and Data Visualisation

A. Class Distribution Analysis

The extreme class imbalance is the defining characteristic of this dataset and has major implications for model selection and evaluation. The pie chart visualisation confirms that 99.9% of transactions are legitimate and just 0.0536% are fraudulent. This means that in the test set of 262,144 samples, only approximately 279 instances belong to the positive (fraud) class.

This imbalance means that accuracy is a misleading metric. A degenerate classifier that predicts all transactions as legitimate achieves 99.9% accuracy with zero fraud detection capability. The focus must therefore shift to precision, recall, F1-score for the minority class, and area under the ROC and Precision-Recall curves.

B. Transaction Type Distribution

Analysis of the transaction type distribution across the full dataset reveals five categories with distinct frequencies. CASH_OUT is the most frequent transaction type (approximately 353,000 transactions), followed by PAYMENT (~319,000), CASH_IN (~213,000), TRANSFER (~79,000), and DEBIT (~7,000).

C. Fraud by Transaction Type

A critical insight from the type-stratified fraud analysis is that fraudulent transactions occur exclusively in TRANSFER and CASH_OUT categories. No fraudulent PAYMENT, DEBIT, or CASH_IN transactions are present in the dataset. This is consistent with real-world fraud patterns where fund exfiltration occurs via transfer to an external account or cash withdrawal. Both fraudulent types appear with approximately equal frequency (~250-260 instances each within the fraud subset).

D. Distribution of Transactional Values

Box plot visualisations of the three continuous features (amount, oldbalanceOrg, newbalanceOrig) reveal strong right-skewed distributions with numerous extreme outliers. The amount feature ranges from near-zero to approximately 10 million units, with the interquartile range concentrated near zero. Similarly, balance features extend to approximately 40 million units. These extreme outliers are characteristic of high-value fraudulent transfers and must be preserved in the training data as they represent genuine signal.

VI. Machine Learning Models

A. Decision Tree Classifier

The Decision Tree (DT) classifier recursively partitions the feature space by selecting splits that maximise information gain measured by entropy. The model was initialised with:

```
classifier = DecisionTreeClassifier(criterion='entropy', random_state=0)
classifier.fit(X_train, y_train)
```

Decision Trees are inherently interpretable and capable of capturing non-linear feature interactions without distributional assumptions. However, they are prone to overfitting on training data, manifesting as overly deep trees that memorise noise. In this study, training and validation accuracy were both observed to be 1.0, suggesting the model's performance on training and validation sets are similar—as reported by the overfitting check module. This is

largely attributed to the highly structured separability of fraudulent transactions (which occur only in specific type codes with characteristic balance-delta patterns).

B. K-Nearest Neighbours (KNN)

The K-Nearest Neighbours classifier classifies a sample based on the majority class among its k closest training instances as measured by a distance metric. The model was configured with:

```
neigh = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=1) neigh.fit(X_train, y_train)
```

With p=1, the Minkowski metric reduces to the Manhattan distance, which has been shown to be more robust to outliers than Euclidean distance (p=2) in high-dimensional financial feature spaces. KNN is a non-parametric lazy learner that defers all computation to query time, making it computationally expensive for large datasets. Training and validation accuracy were both 1.0, indicating consistent performance across both splits.

C. Logistic Regression

Logistic Regression models the posterior probability of class membership as a sigmoid-transformed linear combination of features. The model was instantiated as:

```
classifier = LogisticRegression(random_state=0) classifier.fit(X_train, y_train)
```

Logistic Regression assumes a linear decision boundary in the feature space, which may be insufficient to capture the complex, multi-modal distributions of fraudulent vs. legitimate transactions. Training and validation accuracy were both reported at 0.999, reflecting a stable model that does not exhibit overfitting. However, as seen in the classification report, the model struggles with the minority class due to the linear separator's inability to capture the fraud-region geometry.

D. Random Forest Classifier

Random Forest is a bagging ensemble of Decision Trees, where each tree is trained on a bootstrap sample of the training data and a random subset of features at each split. This introduces diversity among trees and reduces variance compared to a single DT. The model was configured as:

```
classifier = RandomForestClassifier(n_estimators=100, criterion='entropy', random_state=0)
classifier.fit(X_train, y_train)
```

An ensemble of 100 trees was trained with entropy as the splitting criterion. The aggregation of 100 independently trained trees provides robust probability estimates and superior generalisation. Training and validation accuracy were both 1.0, consistent with the structured separability of fraudulent transactions in this dataset.

VII. Results and Comparative Analysis

A. Overall Accuracy

The table below summarises the test-set accuracy of all four models. All models achieve near-perfect accuracy, which is unsurprising given the severe class imbalance and the concentration of fraud in specific transaction types that are readily distinguishable.

Metric	Decision Tree	KNN	Logistic Reg.	Random Forest
Test Accuracy (%)	99.94	99.92	99.92	99.95
ROC-AUC	0.84	0.72	0.61	0.84

AUC-PRC (Train)	1.00	0.82	0.36	1.00
Fraud Precision	0.74	0.71	1.00	0.82
Fraud Recall	0.68	0.44	0.23	0.67
Fraud F1-Score	0.71	0.54	0.37	0.74

Table II: *Comparative Model Performance on Test Set (262,144 samples)*

B. Classification Reports

1) Decision Tree

The Decision Tree achieves a fraud precision of 0.74 and recall of 0.68, resulting in an F1-score of 0.71 for the minority (fraud) class. The model's weighted average precision, recall, and F1 are all 1.00 due to the overwhelming dominance of the non-fraud class (261,865 samples vs 279 fraud samples in the test set).

2) K-Nearest Neighbours

KNN achieves a fraud precision of 0.71 but a notably lower recall of 0.44, yielding an F1-score of 0.54. The lower recall indicates that KNN misses approximately 56% of fraudulent transactions, making it less suitable than tree-based methods for this application. The macro average F1 of 0.77 confirms the degradation in minority class performance.

3) Logistic Regression

Logistic Regression achieves perfect precision of 1.00 for the fraud class (all predicted frauds are actual frauds) but extremely low recall of 0.23 (only 23% of actual frauds are detected). This results in an F1-score of 0.37—the lowest among all models. The high precision/low recall trade-off suggests the linear boundary is overly conservative, rejecting borderline fraudulent transactions. The AUC-PRC of 0.36 on the training set corroborates this finding.

4) Random Forest

Random Forest achieves the best balance between precision (0.82) and recall (0.67) for the fraud class, yielding the highest F1-score of 0.74. The macro average F1 of 0.87 further confirms its superior overall discrimination. Confusion matrix analysis shows 261,824 correct non-fraud predictions, 41 false positives, an undisclosed number of false negatives, and 187 correct fraud detections.

C. ROC Curve Analysis

The Receiver Operating Characteristic (ROC) curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at varying classification thresholds. A perfect classifier achieves AUC = 1.0; a random classifier achieves AUC = 0.5.

Results show that Decision Tree and Random Forest are tied at AUC = 0.84, significantly outperforming KNN (AUC = 0.72) and Logistic Regression (AUC = 0.61). The ROC curves for DT and RF show rapid ascent to high TPR at low FPR, indicating effective ranking of fraudulent transactions. Logistic Regression's linear decision boundary limits its discriminative capacity, as reflected in the lower AUC.

D. Precision-Recall Curve Analysis

For highly imbalanced datasets, the Precision-Recall (PR) curve provides more informative evaluation than the ROC curve. The AUC-PRC measures the area under the precision-recall curve, where a perfect classifier achieves AUC-PRC = 1.0 and a random classifier achieves AUC-PRC equal to the class prevalence (~0.0005 for this dataset).

Decision Tree and Random Forest both achieve AUC-PRC = 1.0 on the training set, demonstrating perfect precision-recall trade-off. KNN achieves AUC-PRC = 0.82, reflecting good but not perfect performance. Logistic Regression achieves only 0.36, consistent with its poor recall. The sharp precision drop at low recall values for KNN and Logistic Regression indicates difficulty in maintaining precision when attempting higher fraud capture rates.

E. Confusion Matrix Analysis

The confusion matrices reveal the following characteristics of model behaviour on the test set:

Model	True Neg (TN)	False Pos (FP)	False Neg (FN)	True Pos (TP)
Decision Tree	261,799	66	~89	~190
KNN	261,814	51	~157	~122
Logistic Regression	261,865	0	216	63
Random Forest	261,824	41	~92	~187

Table III: *Confusion Matrix Summary for All Models (Test Set)*

Notably, Logistic Regression produces zero false positives—every predicted fraud is confirmed—but misses 216 out of 279 fraudulent transactions. In financial fraud detection, false negatives (missed frauds) are typically far more costly than false positives (unnecessary investigation triggers), making Logistic Regression the least suitable model for this use case despite its perfect precision.

F. Calibration Curve Analysis

Calibration curves assess whether a model's predicted probabilities accurately reflect true outcome frequencies. A perfectly calibrated model's curve follows the diagonal reference line (Fraction of Positives = Mean Predicted Probability).

Random Forest demonstrates near-perfect calibration, with the calibration curve closely tracking the ideal diagonal throughout the probability range. This is particularly important for risk-scoring applications where fraud probability estimates are used downstream in risk management systems. Decision Tree shows moderate calibration with slight overconfidence at mid-range probabilities. Logistic Regression exhibits erratic calibration in the 0-0.5 probability range, reflecting its failure to generate well-separated probability estimates for the minority class. KNN shows a generally monotonic but non-linear calibration profile, suggesting systematic miscalibration that could be corrected with isotonic regression post-processing.

VIII. Discussion

The experimental results yield several important insights for the design of real-world financial fraud detection systems:

Model Selection: Random Forest emerges as the recommended model, offering the highest F1-score for the fraud class (0.74), joint-highest ROC-AUC (0.84), best calibration, and strongest AUC-PRC. Its ensemble nature provides robustness to noise and outliers, which are prevalent in financial data.

Linear Model Limitations: Logistic Regression's near-perfect precision but catastrophically low recall (0.23) demonstrates that linear decision boundaries are fundamentally inadequate for this problem. The geometric structure of fraudulent transactions in the feature space is non-linear, requiring the more expressive hypothesis spaces of tree-based models.

Class Imbalance: None of the models employed explicit imbalance-handling techniques such as SMOTE, class weighting, or threshold optimisation. The results suggest that the informative features (particularly transaction type and balance deltas) provide sufficient signal for tree-based models even without resampling. However, applying SMOTE or cost-sensitive learning could further improve minority class recall.

Feature Engineering Opportunity: The creation of derived features—such as balance delta (oldbalanceOrg - newbalanceOrig), amount-to-balance ratio, and binary flags for zero-balance transactions—could significantly enhance model performance by explicitly encoding the patterns characteristic of fraudulent activity.

Operational Considerations: In production deployment, a Random Forest model would require consideration of inference latency (prediction time scales with n_estimators), model serialisation, and periodic retraining schedules to address concept drift as fraud patterns evolve.

IX. Conclusion

This paper presented a comprehensive comparative study of four supervised machine learning classifiers—Decision Tree, K-Nearest Neighbours, Logistic Regression, and Random Forest—for the detection of fraudulent credit and debit card transactions. A complete ML pipeline was implemented, encompassing data ingestion, preprocessing, feature engineering, model training, and multi-metric evaluation.

Key findings are: (1) All models achieve test accuracy exceeding 99.9%, confirming that accuracy alone is insufficient for imbalanced classification. (2) Decision Tree and Random Forest dominate in ROC-AUC (0.84) and AUC-PRC (1.0). (3) Random Forest achieves the best fraud F1-score (0.74) and near-ideal probability calibration, making it the recommended model for deployment. (4) Logistic Regression, despite its high precision, is unsuitable due to its extremely low fraud recall (0.23). (5) Fraudulent transactions occur exclusively in TRANSFER and CASH_OUT categories, providing a strong prior for feature design.

Future work will explore: (i) implementation of advanced imbalance-handling techniques (SMOTE, ADASYN, ensemble resampling); (ii) application of Gradient Boosting (XGBoost, LightGBM) and deep learning architectures; (iii) graph neural network approaches to capture inter-account relationships; (iv) temporal modelling of transaction sequences; and (v) federated learning for privacy-preserving fraud detection across distributed financial institutions.

References

[1] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating Probability with Undersampling for Unbalanced Classification," in Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM), 2015, pp. 159–166.

[2] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," Decision Support Systems, vol. 50, no. 3, pp. 602–613, 2011.

[3] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," Inf. Sci., vol. 479, pp. 448–455, 2019.

[4] J. O. Kim and S. Kim, "Fraud Detection in Financial Transactions Using LSTM," in Proc. Int. Conf. Big Data Artif. Intell. (BDAI), 2021, pp. 112–117.

[5] Y. Liu, X. Zhang, W. Zheng, and Z. Liu, "Graph-Based Fraud Detection in Online Financial Systems," IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 8, pp. 3879–3891, 2022.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[7] L. Breiman, "Random Forests," Mach. Learn., vol. 45, pp. 5–32, 2001.

[8] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Trans. Inf. Theory, vol. 13, no. 1, pp. 21–27, 1967.

[9] L. Rokach and O. Maimon, "Top-down induction of decision trees revisited," in Data Mining and Knowledge Discovery Handbook, Springer, 2005.

[10] E. Lopez-Rojas, A. Elmir, and S. Axelsson, "PaySim: A financial mobile money simulator for fraud detection," in Proc. 28th Eur. Conf. Modelling Simulation (ECMS), 2016.