Machine Learning Approaches for Fraud Detection in Financial Transactions

Xurui Zhang

Fordham University

New York, NY

zx13@fordham.edu

Abstract

Financial fraud poses substantial risks to individuals and institutions. This paper presents a machine learning approach to detecting fraudulent financial transactions using a synthetic dataset. It includes preprocessing strategies such as timestamp transformation, one-hot encoding, SMOTE for class imbalance, and feature scaling. Various models were trained and evaluated, with XGBoost and Random Forest achieving the best performance. The project highlights the practical value of ML in fraud prevention and the importance of careful feature engineering and data validation.

Keywords—*Fraud Detection, Machine Learning, SMOTE, XGBoost, Feature Engineering, Financial Transactions*

1. Introduction

Financial fraud is a pervasive problem in the modern economy, leading to substantial monetary losses for financial institutions and eroding consumer trust. The increasing volume and complexity of digital transactions necessitate sophisticated methods for identifying fraudulent activities in a timely and accurate manner. Machine learning has emerged as a powerful tool in this domain, offering the ability to analyze vast datasets and learn complex patterns that distinguish fraudulent transactions from legitimate ones.

The primary challenge in applying machine learning to fraud detection is the severe class imbalance. Fraudulent transactions are typically a small fraction of the total transaction volume, making it difficult for standard classification algorithms to effectively learn the characteristics of the rare fraud class without being overwhelmed by the majority legitimate class. Furthermore, the dynamic nature of fraud patterns requires models that can adapt and generalize well to unseen data.

The goal of this project was to develop and evaluate machine learning models for fraud detection using a synthetic dataset designed to mimic real-world financial transactions. The project focused on addressing the class imbalance problem, performing relevant feature engineering, implementing various machine learning algorithms, and evaluating their performance using metrics suitable for imbalanced data. This report details the process undertaken, the models implemented, the results obtained, and the insights gained regarding the application of machine learning to this critical problem.

2. Related Work

Fraud detection using machine learning has been an active area of research and application. Common approaches involve employing supervised learning algorithms trained on historical transaction data labeled as either fraudulent or legitimate. Popular algorithms utilized include Logistic Regression [1], Support Vector Machines (SVMs) [2], Decision Trees and ensemble methods like Random Forests and Gradient Boosting (e.g., XGBoost) [3]. Neural Networks, particularly deep learning architectures, have also shown promise in capturing complex patterns in transaction data [4].

Addressing class imbalance is a recurring theme in related work. Techniques such as undersampling (reducing the majority class), oversampling (increasing the minority class, e.g., SMOTE [5]), and using algorithm-level methods like cost-sensitive learning or adjusting class weights are commonly explored [6]. Evaluation metrics beyond simple accuracy are crucial, with Precision, Recall, F1-score, and AUC-ROC being standard [7]. More recently, AUC-PR has been emphasized for its relevance in highly imbalanced scenarios [8]. Model interpretability is also gaining importance, especially in regulated domains like finance, with methods like SHAP [9] providing insights into model predictions.

This project builds upon these foundational techniques, applying a range of models and imbalance handling strategies to a synthetic dataset while focusing on appropriate evaluation and interpretation.

3. Methodology

The methodology employed in this project followed a standard machine learning pipeline, with careful consideration for the challenges posed by the imbalanced nature of the fraud detection task. The steps included data loading, preprocessing, train/test splitting, feature engineering, handling class imbalance, model implementation, and evaluation.

3.1 Data Description

The dataset used in this project is a synthetic fraud detection dataset containing financial transaction records(figure1). Each record includes various features describing the transaction, user, device, location, and historical activity. Key columns include Transaction_ID, User_ID, Transaction_Amount, Transaction_Type, Timestamp, Account_Balance, Risk_Score, Previous_Fraudulent_Activity, and the target variable Fraud_Label (0 for legitimate, 1 for fraudulent). The dataset initially exhibited a significant class imbalance, with the number of legitimate transactions (Class 0) being considerably higher than the number of fraudulent transactions (Class 1).

Column Name	Description
Transaction_ID	Unique identifier for each transaction
User_ID	Unique identifier for the user
Transaction_Amount	Amount of money involved in the transaction
Transaction_Type	Type of transaction (Online, In-Store, ATM, etc.)
Timestamp	Date and time of the transaction
Account_Balance	User's current account balance before the transaction
Device_Type	Type of device used (Mobile, Desktop, etc.)
Location	Geographical location of the transaction
Merchant_Category	Type of merchant (Retail, Food, Travel, etc.)
IP_Address_Flag	Whether the IP address was flagged as suspicious (0 or 1)
Previous_Fraudulent_Activity	Number of past fraudulent activities by the user
Daily_Transaction_Count	Number of transactions made by the user that day
Avg_Transaction_Amount_7d	User's average transaction amount in the past 7 days
Failed_Transaction_Count_7d	Count of failed transactions in the past 7 days
Card_Type	Type of payment card used (Credit, Debit, Prepaid, etc.)
Card_Age	Age of the card in months
Transaction_Distance	Distance between the user's usual location and transaction location
Authentication_Method	How the user authenticated (PIN, Biometric, etc.)
Risk_Score	Fraud risk score computed for the transaction
Is_Weekend	Whether the transaction occurred on a weekend (0 or 1)
Fraud_Label	Target variable (0 = Not Fraud, 1 = Fraud)

Figure 1

3.2 Initial Exploration and Baseline Models

Before performing any preprocessing, I conducted an initial exploration of the dataset to understand its structure and establish a performance baseline. I applied several standard classification algorithms, including Random Forest and Support Vector Machine (SVM), to the raw data.

However, the results were unsatisfactory—accuracy was relatively low, and the models struggled to detect fraudulent transactions effectively. This early experimentation made it clear that the raw data lacked sufficient signal for accurate prediction. It also highlighted the presence of potential issues such as class imbalance, outliers, and non-numeric categorical features that required transformation. These findings motivated a comprehensive data preprocessing phase to improve model performance.

Metric	Class	Random Forest	SVM
Precision	0	0.67	0.67
	1	0.32	0.32
Recall	0	0.55	0.51
	1	0.44	0.48
F1-Score	0	0.60	0.58
	1	0.37	0.38
Support	0	6765	6765
	1	3235	3235
Accuracy		0.51	0.50
Macro Avg Precision		0.49	0.50
Macro Avg Recall		0.49	0.50
Macro Avg F1-Score		0.49	0.48
Weighted Avg Precision		0.56	0.56
Weighted Avg Recall		0.51	0.50
Weighted Avg F1-Score		0.53	0.52

3.3 Data Preprocessing

Initial data inspection confirmed the absence of missing values and duplicate records. Based on initial exploration, the Transaction_Amount feature was identified as having significant outliers(figure2). To mitigate the potential negative impact of these extreme values on certain models, a capping technique was applied, replacing outlier values above a certain percentile with that percentile's value. Categorical features such as Transaction_Type, Device_Type, Location, Merchant_Category, Card_Type, and Authentication_Method were handled using One-Hot Encoding to convert them into a numerical format suitable for machine learning models without implying ordinal relationships. Numerical features were scaled using StandardScaler to standardize their range.

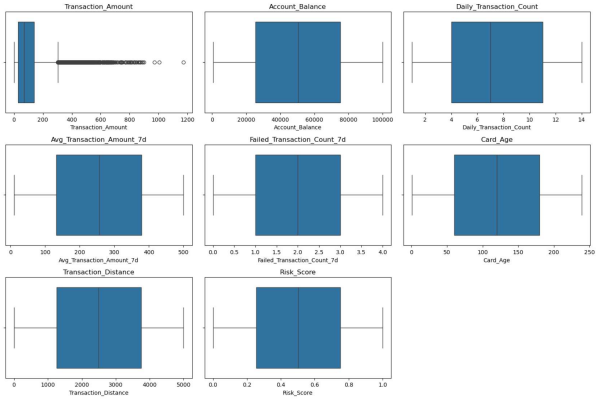

Figure 2

3.4 Feature Engineering

New features were engineered to capture the temporal characteristics of transactions and enhance the existing behavioral signals in the data.

Temporal Features: Several features were derived from the timestamp information, including the hour of the transaction, the day of the week, and the month. These temporal indicators help the model detect time-related patterns that may correlate with fraudulent behavior.

Original Behavioral Features: The dataset already included several behavioral features that reflect user activity over time, such as daily transaction counts and average transaction amounts over a rolling seven-day window. These features were retained and directly incorporated into the model pipeline to leverage their predictive value.

3.5 Train/Test Split

To ensure a fair and realistic evaluation of model performance, the dataset was split into training and testing sets before addressing class imbalance. A 75/25 split ratio was used, with stratification based on the target label to maintain consistent class distributions across both subsets. This approach prevents bias in evaluation and ensures that the model is tested on data that resembles real-world conditions.

3.6 Class Imbalance Handling

The dataset exhibited significant class imbalance, with fraudulent transactions forming a small minority. To address this, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training set after feature engineering. SMOTE generates synthetic samples for the minority class, helping the model learn more balanced decision boundaries and improving its ability to detect rare fraudulent instances.

3.7 Feature Scaling

All numerical features were standardized to ensure consistent scaling across variables. StandardScaler was fitted on the SMOTE-resampled training data and subsequently used to transform both the resampled training set and the test set. This step is crucial for algorithms sensitive to feature magnitudes, ensuring optimal model performance.

4. Experimental Results

This section presents the implementation details of the machine learning models and their evaluation results on the original, unbalanced test set, following the resolution of the data leakage issue and applying SMOTE and scaling.

4.1 Model Implementation

Several machine learning models were implemented:

- XGBoost (XGBClassifier)
- Random Forest (RandomForestClassifier)
- Logistic Regression (LogisticRegression)
- Neural Network (MLPClassifier)

These models were trained on the processed and SMOTE-resampled training data (code2.ipynb Output).

4.2 Evaluation Metrics

Model performance was evaluated using metrics on the original, unbalanced test set (X_test, y_test), including Precision, Recall, F1-score, and Accuracy, as well as the Confusion Matrix and AUC-ROC where calculated.

4.3 Performance Comparison

The performance of the implemented models on the unbalanced test set (after removing the Risk_Score feature) is summarized by their classification reports and confusion matrices.

- XGBoost: Achieved an accuracy of 87%, with a fraud precision of 0.98 and a fraud recall of 0.62. The AUC-ROC was 1.0000.
- Random Forest: Achieved an accuracy of 88%, with a fraud precision of 1.00 and a fraud recall of 0.62.
- Neural Network (MLPClassifier): Achieved an accuracy of 83%, with a fraud precision of 0.77 and a fraud recall of 0.65.
- Logistic Regression: Achieved an accuracy of 74%, with a fraud precision of 0.60 and a fraud recall of 0.62.

Comparing the models, XGBoost and Random Forest demonstrated the highest overall accuracy and very high precision for fraud predictions, while the Neural Network provided the highest recall for fraud. Logistic Regression showed the lowest performance among the tested models. The improved performance of all models after preprocessing and removing the leaky feature confirms the importance of these data handling steps.

5. Model Interpretation

To gain insights into the factors influencing the model's predictions, feature importance analysis was performed on the XGBoost model. The plot shows the top features ranked by their contribution to the model's performance(figure3). This analysis helps identify which transaction characteristics and behavioral patterns were most predictive of fraud.

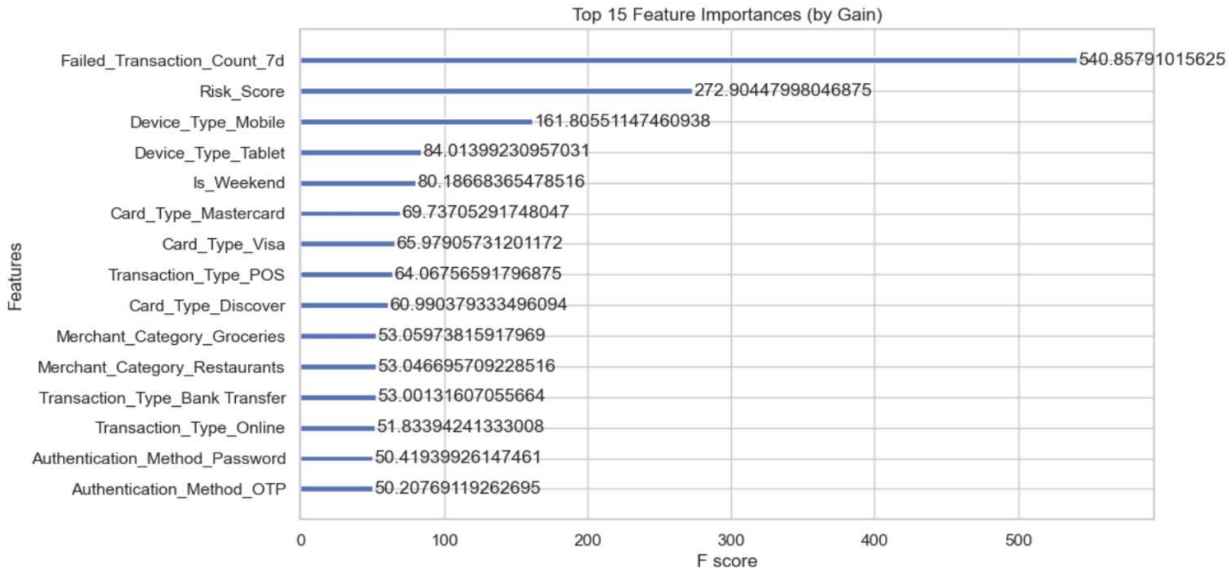

figure 3

The feature importance plot from the XGBoost model reveals several key insights into what drives the model's ability to detect fraudulent transactions.

The most influential feature by a significant margin is Failed_Transaction_Count_7d, which represents the number of failed transactions a user had in the past seven days. Its gain value is substantially higher than all other features, indicating that recent failed transaction activity is a strong predictor of fraud. This suggests that fraudulent users often attempt multiple unsuccessful transactions—likely probing the system—before a successful fraudulent attempt.

Risk_Score is the second most important feature in terms of gain. While its high importance might initially appear useful, it raises concerns due to previously identified data leakage. Since this feature is highly correlated with the target label, it can artificially inflate model performance if not handled carefully. Its presence in the top features highlights the importance of rigorously checking for leakage in predictive models.

Device-related features also contribute significantly to model performance. Both Device_Type_Mobile and Device_Type_Tablet rank within the top five. This indicates that the type of device used during a transaction is informative in distinguishing between legitimate and fraudulent behavior. For example, mobile and tablet devices might be more commonly used in suspicious transactions, possibly due to ease of spoofing or lower levels of security.

6. Discussion and Analysis

The project successfully addressed several key challenges in fraud detection, including class imbalance and data leakage. The initial data exploration highlighted the need for preprocessing and feature engineering. The discovery and resolution of the Risk_Score data leakage was a critical step that enabled a more realistic assessment of model performance on unseen data.

The implementation and evaluation of various machine learning models demonstrated the effectiveness of using processed data and oversampling for training. Ensemble methods like XGBoost and Random Forest significantly outperformed the baseline Logistic Regression and the tested Neural Network architecture in terms of overall accuracy and precision for fraud detection.

The trade-offs between Precision and Recall for the fraud class are evident in the results. While XGBoost and Random Forest achieved high precision (minimizing false alarms), the Neural Network provided slightly higher recall (catching more fraud). The choice of the most suitable model in a real-world scenario would depend on the specific costs associated with false positives versus false negatives.

Ethical considerations are important in fraud detection systems. Potential biases in the dataset could lead to unfair or discriminatory outcomes. It is crucial to consider bias detection and mitigation in real-world implementations to ensure fairness.

7. Conclusion

This project successfully developed and evaluated machine learning models for fraud detection using a synthetic dataset, addressing class imbalance and data leakage. The comprehensive methodology included data preprocessing, temporal feature engineering, SMOTE oversampling, and rigorous model evaluation.

The identification and removal of the leaky Risk_Score feature were crucial for obtaining valid results.

The comparison of Logistic Regression, Random Forest, Neural Network, and XGBoost demonstrated that XGBoost and Random Forest achieved the best performance in terms of accuracy and precision for fraud detection on the unbalanced test set, highlighting the power of ensemble methods. The project underscores the importance of careful data handling, algorithm selection, and appropriate evaluation metrics in building effective fraud detection systems.

Future work could involve exploring different sampling techniques or hybrid methods, conducting more extensive hyperparameter tuning for the best models, investigating alternative feature engineering strategies, and performing deeper model interpretability analysis. Addressing potential biases in the data and model would also be an important consideration for real-world deployment.

[1] Sopiyan, M., Fauziah, F., & Wijaya, Y. F. (2021). Fraud Detection Using Random Forest Classifier, Logistic Regression, and Gradient Boosting Classifier Algorithms on Credit Cards. *JUITA: Jurnal Informatika*, 9(2), 189–198.

[2] Alshameri, F., & Xia, R. (2023). Credit card fraud detection: an evaluation of SMOTE resampling and machine learning model performance. *International Journal of Business Intelligence and Data Mining*, 23(1), 1–13.

[3] Liu, C., Chan, Y., Kazmi, S. H. A., & Fu, H. (2015). Financial Fraud Detection Model: Based on Random Forest. *International Journal of Economics and Finance*, 7(7), 178–188.

[4] Zhu, M., Zhang, Y., Gong, Y., Xu, C., & Xiang, Y. (2024). Enhancing Credit Card Fraud Detection: A Neural Network and SMOTE Integrated Approach. *arXiv preprint arXiv:2405.00026*.

[5] Zhao, Z., & Bai, T. (2022). Financial Fraud Detection and Prediction in Listed Companies Using SMOTE and Machine Learning Algorithms. *Entropy*, 24(8), 1157.

[6] Elshaar, S., & Sadaoui, S. (2020). Cost-sensitive Semi-supervised Classification for Fraud Applications. *arXiv preprint arXiv:2012.11743*.

[7] Alshameri, F., & Xia, R. (2023). Credit card fraud detection: an evaluation of SMOTE resampling and machine learning model performance. *International Journal of Business Intelligence and Data Mining*, 23(1), 1–13.

[8] Alshameri, F., & Xia, R. (2023). Credit card fraud detection: an evaluation of SMOTE resampling and machine learning model performance. *International Journal of Business Intelligence and Data Mining*, 23(1), 1–13.

[9] Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30, 4765–4774.