Suspicious Transaction Detection Using Machine Learning: A Comparative Analysis of Logistic Regression and Random Forest Models

Pramish K.C
https://github.com/vold2emort

Rahul Maharjan
https://github.com/mhrjnrahul

Sushanka Khadka
https://github.com/sushanka-khadka

Samir Paudel
https://github.com/paudelsamir

Abstract

Suspicious Transaction detection is a pressing challenge in the digital era. This paper presents a Machine Learning-driven approach for detecting suspicious transactions. This study utilizes a dataset of 2 million transaction records to develop and evaluate machine learning models for identifying fraudulent activities. We preprocess the data by parsing timestamps, extracting velocity metrics, and encoding categorical variables. Logistic Regression and Random Forest models are trained and compared under unbalanced, balanced weights, and random oversampling (ROS) conditions. By integrating advanced classification algorithms and anomaly detection techniques, our solution offers a dynamic, adaptive fraud detection system that significantly improves accuracy and reduces false positives. Evaluation focuses on accuracy, precision, recall, and F1-score, with recall prioritized to minimize false negatives. The Random Forest model achieves the highest recall and a strong F1-score, making it the preferred choice for fraud detection. This research demonstrates the effectiveness of Machine Learning in identifying hidden patterns and mitigating financial fraud risks. Future work includes refining detection models using real-world transaction datasets and exploring deep learning techniques for improved predictive accuracy.

Keywords: Suspicious Transaction, Random Forest, Logistic Regression, Performance Metrics

Problem Description

The rapid increase of digital transactions has led to a surge in financial fraud, posing significant risks to individuals and institutions. Traditional rule-based fraud detection systems struggle to adapt to evolving fraud patterns, often missing sophisticated attacks. Machine learning offers a promising solution by identifying patterns in transaction data that indicate fraudulent behavior. However, fraud detection datasets are typically imbalanced, with fraudulent transactions being a small minority, which can bias models toward the majority class and result in missed fraud cases (false negatives).

Fraudsters exploit loopholes by performing multiple small transactions under threshold limits, conducting transactions from geographically diverse locations within short time frames, and time of transaction, and variability of amount. Additionally, sudden spikes in transaction volume and frequency can indicate illicit activities, but without advanced detection mechanisms, these patterns may go unnoticed.

This study addresses the challenge of detecting fraud in a transaction dataset while minimizing false negatives. The primary objective is to develop a machine learning model that maximizes recall—ensuring most fraudulent transactions are identified—while maintaining a balanced performance across precision and F1-score. The dataset, synthetic_fraud_data.csv, contains large number of features like transaction amount, merchant details, and timestamps, and a binary target variable (is_fraud). The problem requires preprocessing the data, training models, and selecting the best approach for fraud detection.

Related Works

Fraud detection using machine learning has been extensively studied. Logistic Regression is a popular choice due to its interpretability and computational efficiency (Hosmer et al., 2013). It models the probability of a transaction being fraudulent, making it suitable for binary classification tasks like fraud detection. Random Forest, on the other hand, excels in capturing complex, non-linear patterns in data (Breiman, 2001), often outperforming simpler models in tasks with high-dimensional feature spaces.

Class imbalance is a common challenge in fraud detection, as fraudulent transactions are rare compared to legitimate ones. Bhattacharyya et al. (2011) demonstrated that techniques like oversampling and class weighting can improve model performance by addressing this imbalance. Random oversampling (ROS) duplicates minority class samples, while balanced weights adjust the model's loss function to prioritize the minority class. Recent studies have also explored feature engineering, such as extracting velocity metrics (e.g., transaction frequency), to enhance fraud detection (Dal Pozzolo et al., 2015).

This research builds on these works by comparing Logistic Regression and Random Forest models under different imbalance-handling strategies, using a synthetic dataset to simulate real-world transaction scenarios. The focus on recall aligns with the practical need to minimize missed fraud cases, a priority in financial security.

Proposed Methodology

The proposed methodology involves a systematic approach to Suspicious Transaction detection, encompassing data preprocessing, feature engineering, model training, and evaluation. The workflow is designed to handle large-scale transaction data and address class imbalance, ensuring high recall for suspicious transaction detection.

1. **Data Collection:**

   Since an appropriate dataset for the suspicious transaction is not available a synthetic dataset is used. The synthetic dataset is used for the study. The dataset contains multiple features like transaction amount, merchant details, and timestamps, country, city, card type, and a binary target variable (is_fraud).

2. **Data Preprocessing**:
   During preprocessing, features with timestamp values were processed to capture temporal patterns by transforming them into subsequent features. Features containing dictionary values were expanded into multiple columns to provide more detailed information. Numerical features were scaled using the Min-Max Scaler to ensure unbiased predictions. Categorical features were encoded using various techniques, such as frequency encoding and one-hot encoding, to facilitate model training.

$$Min\ max\ scaler\ (X^{'}) = \frac{X - X_{min}}{X_{max} - X_{min}}.$$

3. **Model Training**:

   The dataset was split into training (80%) and testing (20%) sets. Train Logistic Regression and Random Forest models under three conditions: unbalanced data, balanced weights, and random oversampling (ROS).

   Logistic Regression is a statistical method used for binary classification problems, where the outcome variable y takes one of two possible values, such as 0 or 1, true or false. The purpose of logistic regression is to model the probability of the binary outcome as a function of independent variables. The dependent variable, y, represents the outcome we're trying to predict, while the independent variables, x1, x2…xn, are the features or input variables that influence the outcome.

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n)}}$$

   Random Forest Classifier is an ensemble learning method that combines multiple decision trees to improve classification accuracy. Each decision tree in the forest is trained on a random subset of the data, and during classification, each tree provides a prediction. The final prediction is made by taking the majority vote (mode) from all the individual trees.

Random forests help reduce overfitting by averaging out predictions from multiple models, making them more robust and accurate compared to single decision trees.

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(x)$$

4. **Evaluation**:

Use accuracy, precision, recall, and F1-score as evaluation metrics, prioritizing recall to minimize false negatives. Analyze confusion matrices and ROC curves to understand model behavior. The methodology aims to identify the model configuration that best balances high recall with overall performance, ensuring effective fraud detection in a synthetic transaction environment.

• True Positive (TP): When a positive outcome is accurately predicted by the model, the actual result is also positive.

• True Negative (TN): When a negative result is accurately predicted by the model, the real result is also negative.

• False Positive (FP): When a positive result is predicted by the model but the actual result is negative. Likewise referred to as a Type I mistake.

• False Negative (FN): When a positive result occurs instead of the expected negative one, the model predicted the wrong thing. Likewise referred to as a Type II mistake.

Accuracy: The proportion of correctly predicted instances out of the total instances in the dataset.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: The proportion of true positive predictions out of all positive predictions made by the model.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall: The proportion of true positive predictions out of all actual positive instances in the dataset.

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

F1 Score: The harmonic mean of precision and recall, providing a balanced measure of both.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Implementation Mechanism

The implementation was carried out using Python in a Jupyter Notebook environment, leveraging libraries such as pandas, scikit-learn, and imbalanced-learn. Below is a detailed breakdown of the implementation steps.

1. Data Loading and Preprocessing

A 30% sample of the synthetic dataset was extracted using pandas, reducing computational load while preserving data distribution. The timestamp column was converted to datetime format using pd.to_datetime and split into year, month, day, hour, and minute features. The velocity_last_hour column was parsed using the ast library, converting string dictionaries into a DataFrame with features like num_transactions and total_amount. This step was optimized using .to_list() to minimize memory usage.

Extract velocity metrics from the velocity_last_hour column, stored as a string dictionary, into five features: num_transactions, total_amount, unique_merchants, unique_countries, and max_single_amount. Encode categorical variables (e.g., merchant_category, card_type) using one-hot encoding, ordinal variables (e.g., city_size) with ordinal encoding, and scaling numerical features with MinMaxScaler.

Categorical variables were encoded using OneHotEncoder for nominal features (e.g., merchant_category) and OrdinalEncoder for ordinal features (e.g., city_size). Numerical features were scaled with MinMaxScaler to ensure uniformity. The ColumnTransformer from scikit-learn streamlined the encoding process, producing a final dataset with 31 features, saved as extracted_df.csv for efficiency.

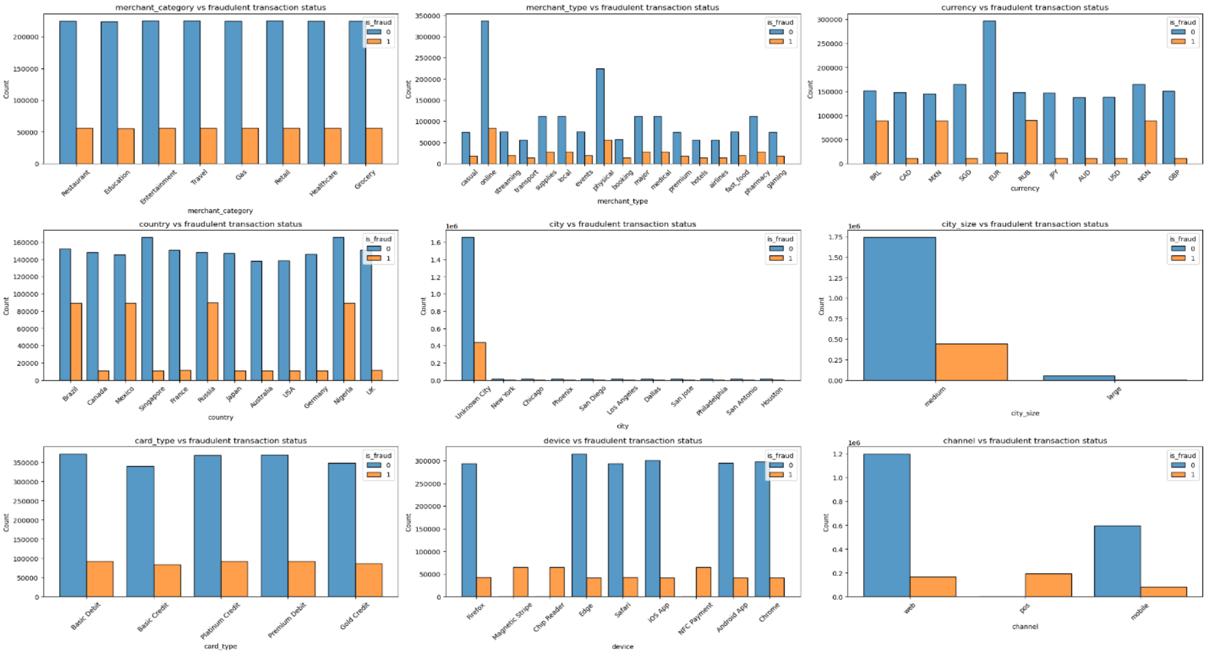

Figure 1. Is_fraud frequency vs. various features

2. **Model Training**

   The dataset was split into training and testing sets using train_test_split (80:20 ratio). Logistic Regression and Random Forest models were trained using scikit-learn's LogisticRegression and RandomForestClassifier, respectively. Three configurations were tested:

   - **Unbalanced**: Default settings, reflecting the natural class imbalance.
   - **Balanced Weights**: Using class_weight='balanced' to prioritize the minority class.
   - **Random Oversampling (ROS)**: Using RandomOverSampler to balance the dataset by duplicating fraud samples.

3. **Evaluation**

   Models were evaluated using accuracy_score, precision_score, recall_score, and f1_score from scikit-learn. Recall was prioritized, as minimizing false negatives is critical in fraud detection. Confusion matrices and ROC curves were generated using Confusion Matrix Display and Roc Curve Display for visual analysis. The Balanced Weights Random Forest Classifier model was selected as the best performer, achieving a test recall of 0.94 and an F1-score of 0.91.

Table 1: Model Performance Metrics

	Train data				**Test Data**			
	accuracy	**precision**	**recall**	**f1**	**accuracy**	**precision**	**recall**	**f1**
Unbalanced logistic	0.93	0.83	0.76	0.79	0.92	0.85	0.76	0.80
Bal weights logistic	0.91	0.67	0.92	0.78	0.91	0.71	0.91	0.80
ROS logistic	0.91	0.67	0.92	0.78	0.91	0.71	0.91	0.80
ROS random forest	0.97	0.87	0.97	0.92	0.96	0.88	0.94	0.91

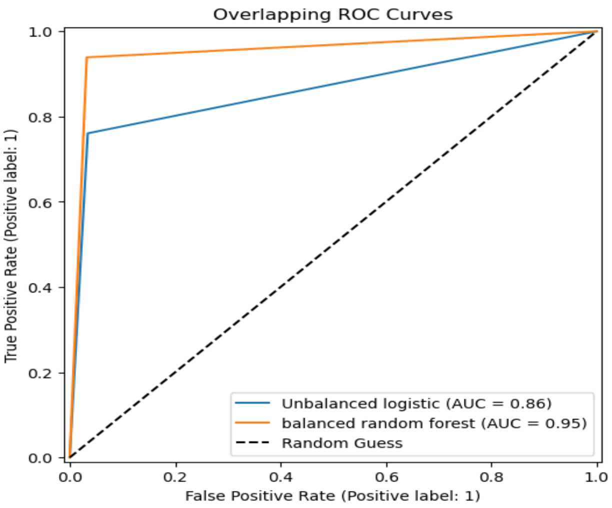

Figure 2. ROC curve comparison

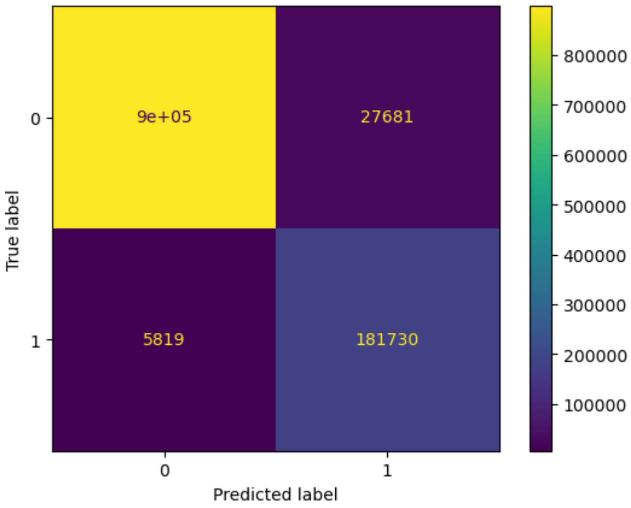

Figure 3. Confusion Matrix of Random Forest Classifier

Future Possible Improvements

While the Balanced Weights Logistic Regression model performs well, several improvements could enhance its effectiveness:

1. **Real-World Data Validation**: The study uses synthetic data, which may not fully capture real-world fraud patterns. Testing the model on actual transaction data could improve generalizability.
2. **Hyperparameter Tuning**: Grid search or random search could optimize model parameters, potentially improving recall and overall performance.
3. **Advanced Techniques**: Ensemble methods (e.g., XGBoost, LightGBM) or deep learning approaches (e.g., neural networks) could capture more complex patterns in fraud behavior.
4. **Feature Expansion**: Incorporating additional features, such as user behavior over longer time periods or network analysis of transactions, could enhance detection accuracy.
5. **Real-Time Deployment**: Implementing the model in a real-time fraud detection system with continuous learning could address evolving fraud tactics.

These improvements could address the limitations of the current study and make the model more robust for practical applications.

References

Paudel, N. (2024) Breast Cancer Prediction: A Comparative Study of Support Vector Machine and Logistic Regression. National College of Computer Studies Research Journal, 3(1), 177-190. https://doi.org/10.3126/nccsrj.v3i1.75469

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602–613. https://doi.org/10.1016/j.dss.2010.08.008

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications, 41(10), 4915–4928. https://doi.org/10.1016/j.eswa.2014.02.026

Hosmer, D. W., Jr., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression. John Wiley & Sons. https://doi.org/10.1002/9781118548387