AI-Based Fraud Detection In Online Transactions

Akansha S. Mogal[1], Rasika T. Mogal[2], Rajashri S. Rahane[3]
[1,2]*PG Student, K.T.H.M. College, Nashik, Maharashtra, India*
[3]*Assistant Professor, Department of Computer Science, MVP Samaj's S. V. K. T. College, Deolali Camp, Nashik, Maharashtra, India*
doi.org/10.64643/IJIRT211714

***Abstract*—Due to digitisation of payments, it is easier to make transactions, shop and conduct financial activity. However, within digital transactions there is a requirement to detect fraudulent activity with a high degree of accuracy. In this paper, we have implemented a Machine Learning method to detect potential fraud in online transactions. The Indian Credit Card Fraud Detection Dataset was used which consists of 500000 synthetic transactions with 18 features. Within the dataset, there are only 9,926 fraudulent transactions (1.9852%), compared to 490074 Genuine transactions (98.0148%) indicating high class imbalance within data. Models were trained and tested on the data using an 80-20 train test split. Logistic Regression had an accuracy of 97.596%, precision of 44.80%, recall of 91.48%, F1 score of 60.15%, PR-AUC of 85.18%. Random Forest had an accuracy of 98.514%, precision of 58.04%, recall of 90.42%, F1 score of 70.70%, PR-AUC of 90.02%. On balance, Random Forest outperformed Logistic Regression with fewer false positives. However, Logistic Regression had a higher recall. These results were obtained on this specific synthetic dataset and may not be representative of a real-world implementation within banking.**

***Index Terms*—Artificial Intelligence, Machine Learning, Online Payment Security, Fraud Detection, Anomaly Detection, Predictive Modelling, Digital Banking, Financial Data Analysis, Supervised Learning.**

I. INTRODUCTION

Payment services in online environments are very common. They allow customers to make purchases, pay bills and send money to other people via UPI, Debit/Credit cards, Net Banking and Digital Wallets. However, with such an increase in speed and volume of digital transactions, efficient methods of detecting fraud within these transactions are necessary. Using automated techniques can provide insight into which data points require investigation. In situations where fraudulent and valid transactions are known, fraud detection can be considered a supervised classification problem. The machine learning model will learn patterns in given data features to predict if a transaction is fraudulent or not. Features within the data may include value of transaction, time, type of merchant, payment method, geographical information and customer profile. The performance of the model will depend on many variables of the data, including feature selection, any preprocessing, class distribution and evaluation used. If only a small proportion of transactions in the data set are fraudulent, accuracy may not be the best measure of model performance. In this research, both Logistic Regression and Random Forest models are implemented and compared. Logistic Regression is a probabilistic model that can be used as a baseline whereas Random Forest uses an ensemble of decision trees to capture non-linear relationships. This research has been conducted on synthetic data, so should be interpreted with this in mind.

A. Problem Statement

If fraudulent transactions are not detected, money may be lost. However, if a legitimate transaction is incorrectly labelled as fraudulent, this can cause inconvenience to customers and extra work for review teams. Another issue with detecting fraudulent transactions is the imbalance within the data. Only 1.9852% of observations were fraudulent. Thus, an efficient system is needed to identify fraudulent transactions, without creating too many false alarms.

B. Aim

To apply two supervised machine learning algorithms (Logistic Regression and Random Forest), and compare their performance at detecting potential fraudulent transactions within an imbalanced dataset of transactions occurring in India.

C. Objectives

- To study how supervised ML algorithms can be applied to detect fraudulent online transactions.
- To check the dataset of transactions in India, and ensure that the attributes and target variable contained within are correct.
- To preprocess the transaction attributes to be in a standardised format suitable for training and predicting with models.
- To apply Logistic Regression and Random Forest models using Python and Scikit-learn.
- To compare the model’s using accuracy, precision, recall, F1-score, PR-AUC and confusion matrix metrics.

II. LITERATURE REVIEW

A simulated dataset was created by Lopez-Rojas, Elmir and Axelsson to investigate fraud detection in mobile payments [1] . This can be useful as real data may be hard to access due to privacy concerns. However, real payment networks may not be accurately represented by simulated datasets.

Tekkali and Natarajan investigated fraud detection in electronic payment systems. They found that balancing data, feature selection and using supervised learning models were important factors. It was highlighted that fraud detection can be challenging due to the imbalanced nature of the class labels [2] . In fraud datasets, fraudulent examples may be vastly outnumbered by non-fraudulent examples. Therefore, during model training, predictions could be heavily biased towards the majority class. To overcome this, data balancing techniques and feature selection can be applied. However, it is important to consider how effective these techniques are by testing the model on an appropriate test dataset.

Kamble et al. investigated fraud detection in UPI payments using behavioural analysis and stacked generalisation [3] . From this it can be seen that contextual information about UPI transactions in India is very useful in detecting fraud. It may not just be the value within a payment which determines if it is fraudulent, but also factors such as the type of payment, type of merchant and contextual information.

Vijayanand and Smrithy investigated explainable ensembles for detecting fraud in mobile payments. It was highlighted that having explainable models in this case is very beneficial [4] . In a financial case, knowing why a transaction received a specific risk score can allow analysts to gain more insight from the model’s predictions. Having explainable models does not necessarily mean that they are accurate and fair however.

Reddy et al. investigated fraud detection in online payments using supervised machine learning methods [7] . Transaction patterns were considered to differentiate between genuine payments and suspicious activity. It was found that by using ensemble models, more reliable fraud detection can be achieved than with simple linear models in complex transaction patterns.

Almazroi and Ayub propose a model to detect fraud in online payments using machine learning [8] . Class imbalance was addressed within this paper, and optimal feature representation was considered. It was found that tree and ensemble models had high levels of precision and F1 scores and reduced false positives in digital payments.

Aladakatti et al. compared different machine learning models to detect fraud in online payments [9] . A trade-off was found between precision and recall depending on the classifier. To evaluate classifiers on skewed financial data, metrics other than accuracy must be considered.

Chandana et al. developed a framework to detect fraud in financial transactions. Different classifiers were considered including logistic regression and Support Vector Machines, based on key transaction and behaviour features. These features included value, time, location and spending patterns [10] . Within this study it was shown that behaviour was very important to ensure secure financial transactions in digital payments.

Shah et al. investigated detecting fraudulent transactions in online payments using machine learning methods and deep learning methods [11] . Different models were compared, including a binary classifier (Logistic regression) and neural networks. Using confusion matrix metrics, it was shown that pre-processing and comparing models allows for effective anomaly detection.

Dodda investigated detecting fraud in real-time digital

payments [12] . Low latency scoring, transaction fingerprints and adaptable classification were investigated to reduce financial loss without affecting genuine customers.

Various methods have been investigated to detect fraud in payments. These include simulated data, supervised learning, balancing data, contextual data and ensembles [1] – [4] and [7] – [12] . Furthermore, it is clear that a single metric should not be used to measure how well a model detects fraud. If the positive class is rare, precision, recall, F1 score and precision recall analysis can be extremely useful. Within this investigation, 2 commonly used supervised models were compared to detect fraud in a synthetic dataset of Indian payments. This does not mean that only these methods have been used, as discussed in the literature.

III. DATASET DESCRIPTION

Data has been taken from the Indian Credit Card Fraud Detection Dataset on Kaggle [5] . The data consists of 500,000 examples across 18 features. Within the dataset description, it was specified that this is synthetically created data for academic purposes. Therefore, the transaction names, card information and other personal-like data within the dataset does not belong to real customers.

The data spans from 2023-01-01 to 2024-12-30. The target variable, is_fraud, takes values of 0 or 1, indicating normal or fraudulent transactions respectively. The number of examples in each class were checked from the CSV file. There are a total of 490,074 normal transactions (98.0148%), and 9,926 fraudulent transactions (1.9852%). Due to this imbalance, there are significantly more examples in the normal class, compared to the fraudulent class. The 18 features include: date and time of transaction, synthetic card holder and card IDs, age, occupation, state, merchant, category, payment type, value, transaction ID, whether the transaction was fraudulent and reason for fraud.

The target variable is included in the data for supervised learning. Features which contain directly identifiable information have been removed meaning that the model will not rely on specific values for individual data points to form meaningful patterns for predicting new transactions.

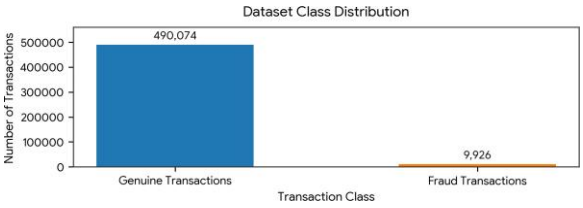


Fig. 1. Class distribution of the Indian transaction dataset.

Fig. 1 shows distribution of classes within the Indian transaction dataset. Within the complete dataset, it can be seen that there are far fewer fraudulent examples than normal ones. 490,074 examples are normal, and only 9,926 examples are fraudulent. This means that 98.0148% of examples in the dataset are normal, and only 1.9852% of examples are labelled as fraudulent. As there are significantly fewer examples in the fraudulent class, accuracy must be considered alongside precision, recall, F1-score and PR-AUC.

IV. METHODOLOGY

Loading the data, pre-processing, training, evaluating and storing the model was completed. Pandas was used for data manipulation, Scikit-learn for pre-processing and classification [6] . Joblib was used to store objects. Both classification models were trained and tested on the same data so that results can be directly compared.

A. Data Pre-processing

The data was loaded from CSV format. From this, the target variable was separated from the input variables. Input categorical variables (Occupation, State, Merchant, Category, Payment Method) required encoding. This converts categorical data into numeric data which can be used by ML models. Input numerical variables (Age, Value, Hour, Day of Week, Month) were pre-processed. To ensure the pre-processing steps are the same when training and evaluating the models a pre-processing object was created.

B. Evaluation

Results can be directly compared from the stored outputs as both models were trained and tested using the same split. The data was divided into 80% for training and 20% for evaluation. As both models were evaluated on the same test data, the results can be compared consistently. 100,000 instances were used

within the confusion matrix in testing. Using the same test data ensures that the comparison between models is fair, and differences in output are due to differences in the algorithm rather than differences in testing data.

C. Logistic Regression

Binary classification can be completed using Logistic Regression. For this problem the labels are 'genuine' and 'fraud'. The input variables are summed with weights applied, then passed through the logistic function to give a probability between 0 and 1. Once the probability is calculated, a decision boundary can be applied to assign a class label.

The probability model can be defined as:

$$P(y=1|x) = 1 / (1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)})$$

Where x represents the input variables, β values represent the learned coefficients of the model and y=1 represents fraud. As Logistic Regression is a simple model it is useful as a baseline. It is fairly interpretable, and outputs probabilities.

D. Random Forest

This method uses multiple decision trees using different samples of data and subsets of inputs. Within each decision tree, decision rules have been learnt. Multiple trees are used to make a prediction meaning complex patterns and interactions in the transaction data can be captured. Having multiple trees produce the prediction reduces dependency on any single tree meaning the overall prediction will be more robust.

V. PERFORMANCE EVALUATION METRICS

Accuracy measures all correctly predicted instances. However, accuracy may be misleading in imbalanced datasets. Precision measures the proportion of predicted fraudulent instances that actually are fraudulent. This reduces the number of unnecessary investigations. Recall measures the proportion of fraudulent instances that were correctly identified. This reduces the number of unseen fraudulent instances. The F1 score is the harmonic mean between precision and recall. The PR-AUC (average precision saved) summarises the trade-off between precision and recall across different thresholds and also focuses on the rare positive class.

From the confusion matrix, we can find the four values: True Negative (TN), False Positive (FP), False Negative (FN) and True Positive (TP).

Precision = TP / (TP + FP) | Recall = TP / (TP + FN) | F1 = 2 × (Precision × Recall) / (Precision + Recall)

VI. RESULTS AND DISCUSSION

Random Forest had the best performance in terms of accuracy, precision, F1-score and PR-AUC. Logistic Regression model had a marginally better recall. In this experiment, the Random Forest has given a better balance between fraud and accurate alerts.

Model	Accuracy	Precision	Recall	F1-score	PR-AUC
Logistic Regression	97.60%	44.80%	91.48%	60.15%	85.18%
Random Forest	98.51%	58.04%	90.42%	70.70%	90.02%

Table 1 shows that Random Forest had an accuracy of 98.514%, a precision of 58.04%, F1 score of 70.70% and PR-AUC value of 90.02%. Logistic Regression had an accuracy of 97.596%, precision of 44.80%, F1-score of 60.15% and PR-AUC value of 85.18%. Logistic Regression model had higher recall at 91.48% compared to the Random Forest model which had a recall of 90.42%.

Looking at the confusion matrices for each model: Logistic Regression had 95782 true negatives, 2235 false positives, 169 false negatives, and 1814 true positives. Random Forest had 96721 true negatives, 1296 false positives, 190 false negatives and 1793 true positives. Random Forest has less false positives than Logistic Regression.

However, Logistic Regression was able to detect a smaller amount of fraud cases. Therefore, depending on how much we are willing to pay for extra false alarms, and not detecting fraud can determine which model is more suitable.

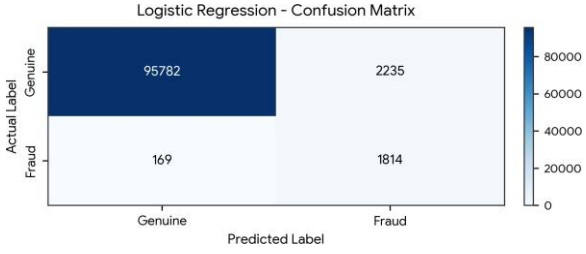


Fig. 2. Confusion matrix for Logistic Regression.

From Figure 2, we can see 95,782 true negatives and 1,814 true positives were correctly identified by the Logistic Regression. 2,235 false positives and 169 false negatives were produced. The small number of false negatives matches the improved recall in comparison to other models. However, there was a large number of false positives meaning that some genuine transactions have been highlighted for further investigation.

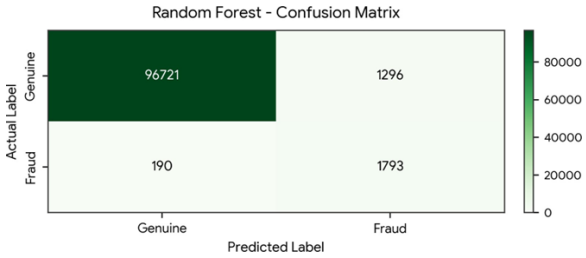


Fig. 3. Confusion matrix for Random Forest.

As shown in Figure 3, there are fewer false positive results for the Random Forest model than Logistic Regression. Specifically, there were 1,296 false positives and 190 false negatives. However, there were a slightly increased number of frauds that went undetected compared with Logistic Regression.

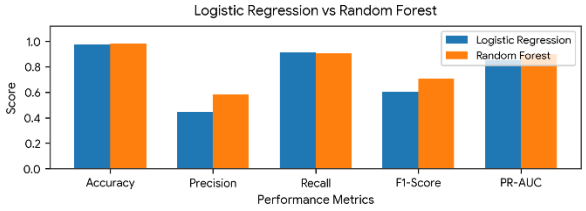


Fig. 4. Comparison of Logistic Regression and Random Forest performance.

The graphs above (Fig. 4) compare model performance in terms of Accuracy, Precision, Recall, F1-Score and PR-AUC. In most cases, Random Forest outperforms Logistic Regression, however, Logistic Regression has a slight edge in Recall. Overall, Random Forest is considered the most effective model in this test case. However, in this case, recall should also be considered.

VII. FUTURE SCOPE

To improve further upon this research, we could:

- Test models against real bank transaction data (with appropriate permissions/privacy).
- Validate models across time to ensure models perform well on transactions from future times/ different payment sources.
- Try other ensembling/deep learning models such as Gradient Boosting (XGBoost, LightGBM) or Neural Networks.
- Explore how we may choose optimal probability thresholds within the model to suit the cost of false positives and false negatives in different scenarios.
- Add interpretability methods (SHAP/LIME) to allow financial analysts to understand which transaction variables have caused a given score in fraud risk.

VIII. LIMITATION OF STUDY

- As this data is synthetically generated, it may not reflect real-world scenarios in banking payments and fraudulent activity.
- The results shown are for an 80/20 split of the given data, and therefore should not be considered as validated deployment within a real banking scenario.
- The model performance will vary depending on feature selection, pre-processing and default classification threshold.
- When deployed in real-world scenarios, strong measures must be in place to protect user data privacy and sensitive financial data must be appropriately secured.
- Manual checks must also be implemented and models must be regularly monitored to account for shifting fraudulent activity.

IX. CONCLUSION

Logistic Regression and Random Forest models were applied to detect fraudulent data within a synthetic Indian dataset containing 500,000 records. Random Forest gave improved results overall with an accuracy of 98.514%, precision of 58.04%, F1 score of 70.70% and PR-AUC of 90.02%. Fewer false positive alerts were detected with the Random Forest model. Logistic Regression had a slightly higher recall (91.48%) compared to Random Forest (90.42%) meaning fewer fraud cases were missed. Overall, Random Forest performed better according to the majority of metrics. However, depending on if it is more important to reduce missed fraud instances or reduce unnecessary alerts within this application, the preferred model can change.

REFERENCES

[1] E. A. Lopez-Rojas, A. Elmir, and S. Axelsson, “PaySim: A financial mobile money simulator for fraud detection,” in *Proc. 28th European Modeling and Simulation Symposium (EMSS)*, 2016, pp. 249–252.

[2] C. G. Tekkali and K. Natarajan, “Smart Fraud Detection in E-Transactions Using Synthetic Minority Oversampling and Binary Harris Hawks Optimization,” *Computers, Materials & Continua*, vol. 75, no. 2, pp. 3171–3187, 2023, doi: 10.32604/cmc.2023.036865.

[3] V. B. Kamble, K. Pisal, P. Vaidya, and S. Gaikwad, “Enhancing UPI Fraud Detection: A Machine Learning Approach Using Stacked Generalization,” *International Journal of Multidisciplinary on Science and Management*, vol. 2, no. 1, pp. 69–83, 2025, doi: 10.71141/30485037/V2I1P108.

[4] D. Vijayanand and G. S. Smrithy, “Explainable AI-enhanced ensemble learning for financial fraud detection in mobile money transactions,” *Intelligent Decision Technologies*, vol. 19, no. 1, pp. 52–67, 2025, doi: 10.1177/18724981241289751.

[5] K. Sharma, “Indian Credit Card Fraud Detection Dataset,” *Kaggle*, 2026. [Online]. Available: https://www.kaggle.com/datasets/quantumkaran/indian-credit-card-fraud-detection-dataset. Accessed: Nov. 20, 2026.

[6] Scikit-learn Developers, “Scikit-learn: Machine Learning in Python,” *Official Documentation*. [Online]. Available: https://scikit-learn.org/stable/. Accessed: Nov. 20, 2026.

[7] M. R. Reddy, H. Q. Owaied, M. M. Abdulhasan, P. Karthik, M. Rajkumar, and P. U. Kumar, “Fraud Detection of Online Transaction Using Machine Learning,” in *Proc. 2024 Int. Conf. Augmented Reality, Intelligent Systems, and Industrial Automation (ARIIA)*, 2024, doi: 10.1109/ARIIA63345.2024.11051851.

[8] A. A. Almazroi and N. Ayub, “Online Payment Fraud Detection Model Using Machine Learning Techniques,” *IEEE Access*, vol. 11, pp. 137188–137203, 2023, doi: 10.1109/ACCESS.2023.3339226.

[9] D. Aladakatti, G. P., A. Kodipalli, and S. Kamal, “Fraud Detection in Online Payment Transaction Using Machine Learning Algorithms,” in *Proc. 2022 Int. Conf. Smart and Sustainable Technologies in Energy and Power Sectors (SSTEPS)*, 2022, pp. 223–228, doi: 10.1109/SSTEPS57475.2022.00063.

[10] S. Chandana Y., R. Devi D. P., N. J. Joy, and N. S. Kumar, “Financial Fraud Detection,” *IJIREEICE*, vol. 13, no. 11, pp. 45–50, Nov. 2025, doi: 10.17148/IJIREEICE.2025.131108.

[11] N. P. Shah, K. B. Panchal, V. A. Jambhale, G. K. Kharat, and S. N. Galinde, “Online Fraudulent Transaction Detection Through Machine Learning and Deep Learning Algorithms,” *IJIREEICE*, vol. 12, no. 5, pp. 142–147, May 2024, doi: 10.17148/IJIREEICE.2024.12524.

[12] A. Dodda, “Real-Time Fraud Detection in Digital Payments Using Machine Learning and Big Data Analytics,” *IJIREEICE*, vol. 8, no. 12, pp. 30–35, Dec. 2020, doi: 10.17148/IJIREEICE.2020.81206.