Predictive Modeling for Fraudulent Credit Card Transactions

Sadish Sendil Murugaraj[1*], G. Swetha[2], M. Bhargavi[3] and Y. Akhila Sirisha[4]
{drsadishsendilm@veltech.edu.in[1] , vtu19179@veltech.edu.in[2] , vtu19258@veltech.edu.in[3],
vtu19564@veltech.edu.in[4]}

Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai – 600062, Tamil Nadu, India[1, 2, 3, 4]

Abstract. Currently, Credit card fraud is a big issue in the recent financial systems, causing huge financial losses on the consumer and the organization in the financial system. Thus, in order to eliminate the aspect of the same misuse and risk of financial loses by unauthorized transactions, good fraud detection systems should be very fast and very precise. To compare this several machine learning models such as Logistic Regression, Decision Trees, Random Forest, and XGBoost Naive Bayes, Random Forest does the best job among them due to the strong classification accuracy, rebalance capability against the rare class, and insensitiveness with respect to over fitting. The fraud detection workflow will be done in some steps which are Data acquisition, Preprocessing step, Feature creation step, Model construction, and last but not least Model deployment. As the fraudulent transactions count is greatly outnumbering the genuine transactions, it is highly recommended to balance the SMOTE, the dataset as such that the model can detect the fraud patterns very well. Furthermore, Feature scaling can also be made for the sake of the stability and dependability of the model. Then, the trained Random Forest model can be cast through Flask Web API which will be able to detect the fraud in real time in order to classify the transaction as a fraud or not at one spam time. As shown in the Figure the Random Forest model has remarkably outperformed both accuracy and AUC-ROC benchmarks, i.e.98% and 96.9%, respectively compared to each of the machine learning model. This research will extend the literature of fraud detection with the comparison of Random Forest and data mining model instead. It has a level instant scalable significant fraud detection system it will be good to make the transaction safer thereby reducing the outflow of money.

Keywords: Credit card fraud, Machine Learning, Random Forest, SMOTE, Flask API.

1 Introduction

Credit card frauds are now a popular type because of the rapidly increasing financial loss to both individual and bank [2]. In the digital world of today, the secure payment procedures may be attacked by criminals using stolen card details, phishing, skimming and identity theft [3]. Among the most prevalent and sophistical forms of credit card fraud is card not present (CNP) frauds, where the physical card is not needed for online purchases [6]. Postal fraud is difficult to identify as there are no mechanisms to directly validate the authenticity of the card entry [7].

The class imbalance problem in transaction logs (negative instances are in the top of 0.5%) also makes fraud detection systems encounter extreme difficulties. This imbalance can cause bias in the machine learning algorithms and decrease their capability for accurately recognizing the fraudulent transactions [10,11]. Further, adversaries continually change their behaviors, making a dynamic and adaptable evolution of detection systems as an inevitable procedure [12].

ICITSM-Part II 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2358061

To overcome these challenges, several artificial intelligence and machine learning techniques have been proposed to detect fraud. Deep learning and hybrid models are promising, but ensemble methods such as Random Forest performed well in precision, flexibility, and data imbalance [15]. In this paper, we study AI-based fraud detection, examining several models and settling on Random Forest as the best. Well, the system is trained on a synthetic dataset generated through applying the SMOTE oversampling [8] and scaling features to improve computational times. We hope to build intelligent fraud detection models that provide immediate feedback Flask Web API provides predictive alerts and live transaction classification on-the-fly.

2 Related works

Fraud detection is one such challenging task because the transaction data sets are highly imbalanced and the majority class is heavily out-numbered by the fraudulent transactions. Abdulla Muaz et al. compared the performance of SMOTE, RUS and DBSMOTE with ANN, Gradient Boosting, Random Forest along with Stacked enabled models. Results suggested that SMOTE combined with Random Forest performed best in fraud detection with higher recall and precision [1]. Fawaz Khaled Alarfaj et al. ML and DL models were used to improve credit card fraud detection. They were compared to Random Forest, SVM and Logistic regression models, i.e. CNN. According to their findings, CNN-based models achieved high accuracy, in particular with an AUC score of 98% [4].

A self-adapting and efficient dandelion algorithm for fraud detection-based feature selection was proposed by Honghao Zhu et al. Current fraud detection techniques are inadequate at large and skew data sets as it is hard to classify accurately the fraud and normal transactions. SEDA is an enhanced version of a classical Dandelion Algorithm (DA), that modifies it to avoid redundant parameters and insert an adaptive seeding radius based on the idea of swarm-SAPT to increase the efficiency. Applying it to credit card fraud detection demonstrated more efficient classification performance, AUC and G-mean values are higher and the computational cost is lower. [5] Jashandeep Singh et al. address all factors of fraud detection, namely we want to reduce the false positives while ensuring that they are not dismissed through the monitoring of transaction flows. They were interested in real-time fraud detection with online banking and oscillated between many machine learning models. They also suggested a holistic solution by relying on the anomaly detection, deep reinforcement learning, financial literacy for a sophisticated fraud prevention in banking transactions.

Dang et al. were interested in the area of machine learning applied to credit card fraud detection, in particular in the approach for handling the issue of data imbalance. Since the number of transactions fraudulent is much less than the regular ones, they utilized resampling techniques like SMOTE, ADASYN to create a more balanced dataset before building machine learning models such as Logistic Regression, Random Forest, K Nearest Neighbors (KNN) and Decision Tree. They found that the models trained with synthetic data were substantially more accurate than the rest, achieving an overall 99% performance. DRL is not robust with imbalanced data, and gets much lower accuracy (Table II). This article draws attention to the importance of implementing resampling techniques, increasing efficiency and accuracy of all ML-based fraud detection applications. With the growing popularity of online payments, detecting fraudulent credit card transactions has been increasingly more important [13].

Thanh Cong Tran and Tran Khanh Dang and others reviewed some machine learning techniques to address one of the more challenging fraud detection challenges—imbalanced data sets. Due to model upgraded performance, they used SMOTE to balance the data set before testing it with classification models such as Random forest, Decision tree, K Nearest Neighbour algorithm, and Logistic Regression models. This finding indicates the effectiveness of Random Forest and the KNN in identifying the fraudulent transactions with perfect accuracy. They also witnessed the downside: fraud detection in the real world remains a problem. A large number of breakthroughs then need to be made for fraud detection systems capable of processing high volumes of transactions quickly and economically [14]. Yu Xie et al. introduced the TbHGN model for fraud detection under transactional behavior. The typical shortcoming of legacy fraud detection models is their incapability of recognizing important transaction attributes and spending behavior patterns. TbHGN tackles this problem by adopting dedicated feature extraction module on the important details of transactions and a behavioral analysis module on the normal spending behavior [16].

3 Methodology

Fig 1 illustrates Random Forest-based fraud detection process. It starts with transaction data preprocessing, including feature scaling and SMOTE to handle imbalanced datasets. The model predicts and notifies the users that the transaction is legitimate or fraudulent using the real time fraud classification model after the preprocessing. The steps involved are data augmentation, dataset splitting, and applying ML models. Based on the performance analysis. Random Forest is chosen for further processing. Finally, fraud classification is performed using majority voting from multiple decision trees to enhance accuracy. This model is used to gain accuracy and efficacy in the identification of fraudulent transactions.

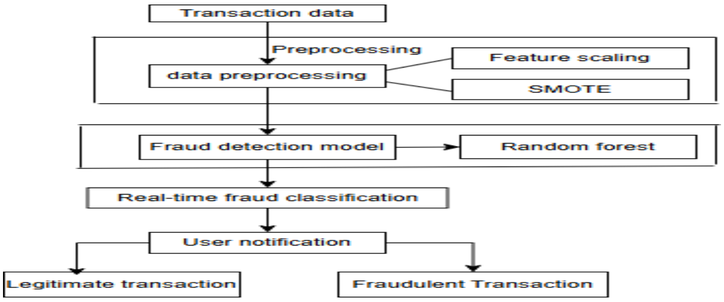

Fig. 1. Research Process of Random Forest Model.

Out of the mandatory fields transaction amount, city, time, store and labels, a fraud prediction is made. Data prepare starts with a data clean then over to 'day prep' for analysis. IT handles missing values, null values, encoding the categorical data in proper format. And, normalize to make all of those features into a similar format for one another.

Because the number of fraudulent transactions is the minority, we utilize technologies, for example, SMOTE and under sampling, to avoid bias dominating and improve the model on detecting fraud properly. After the preprocessing of features, the most discriminative features are selected in order to increase the modeling prediction ability. Subsequently, a Random Forest (a decision trees ensemble to navigate through different dataset sections and to classify transactions) is employed. At last, the final decision is obtained by majority voting, thus ensuring the robustness for the fraud detection system. Furthermore, to put it in perspective, accuracy, precision and recall score and AUC-ROC value are all in the comfort range of freedom for the detection of fraudulent transactions with very low false positive detections. Last, we deploy the trained model as a real-time fraud detection system by using a Flask API which supports easy integration with banking systems to do the real-time checking of transactions, and suspicious activity detection allowing the bank to have its own updated fraud detection system.

Equation (1) is the transaction dataset containing 284,000 samples and 480 fraudulent ones.

$$T_{transactions} = \cup \sum_{i=1}^{284807} T_i \tag{1}$$

Where Ti is a specific credit card transaction, as in Equation (2).

$$T_i = \begin{bmatrix} V_1 \ldots. & V_{29} & V_{30} \\ Time \ldots & Amount & Class \end{bmatrix} \tag{2}$$

The dataset has a high degree of imbalance, and the resampling techniques like the Synthetic Minority Over- Sampling Technique (SMOTE) have to be used for improving fraud detection performance. The mathematical expression of the SMOTE balancing technique is given in Equations (3) and (4).

$$x'_{smote} = x_{minority} + \lambda(x_{nearest} - x_{minority}) \tag{3}$$

Where λ is a randomly selected value between 0 and 1, utilized to enable synthetic sample generation.

$$Y_{balanced} = Y_{original} \cup Y'_{smote} \tag{4}$$

The resampled dataset is then used for feature scaling with StandardScaler normalization, as shown in Equation (5).

$$x'_{scaled} = \frac{x_{original - \mu}}{\sigma} \tag{5}$$

Here μ is the meaning of each feature and σ is its standard deviation.

3.1 Feature Selection

For improving model efficiency, relevant and important features from V1 to V30 are selected based on their impact on fraud detection, ensuring the model focuses on important transaction attributes, and improving performance.

3.2 Random Forest Model Training

Random Forest is in fact an ensemble learning algorithm. It leverages multiple decision trees to improve classification and mitigate overfitting. Random Forest Instead of using just one decision tree, Random Forest uses bagging, i.e. a bunch of them, and takes the majority vote of all the trees for classification. It is useful in the fraud transaction detection, as it can find the complex relationships of the transaction data. The Random Forest algorithm follows:

1. Bootstrapped Sampling (Bagging Technique)

Rather than training on the full dataset, this procedure selects random subsets of data with replacement. This is done in order to enhance the model's capacity to generalize because each decision tree can learn from different distributions of data.

2. Feature Randomization at Each Split

Instead of examining all the features at each split, a randomly chosen subset (e.g., V1–V30) is employed. This strengthens the model and avoids decision trees from similarity, resulting in improved overall performance.

3. Training Decision Trees

The decision trees are one of the most popular methods for machine learning that can be applied to both classification and regression problems. They work by dividing the data to smaller sets based on different criteria, also known as decision making. Each split makes a tree based decision structure; the decision rules are the branches and the nodes are either some feature-transaction value pair or an observation. The leaves at the end output the prediction of the result. Decision trees make it easy to directly graph and read how decisions are being made. To create a good decision tree:

- Calculate node purity by entropy or Gini impurity.
- Choose the best feature to split on based on Information Gain.
- Recursively partition and partition again as long as specific conditions are fulfilled, i.e., achieving an optimal number of samples per node or maximum tree depth.

While splitting nodes, Gini Impurity is calculated using equation (6):

$$G_{ini} = 1 - \sum_{i=1}^{C} \rho_i{}^2 \tag{6}$$

Where pi is the probability of a class (genuine or fraud), and C is the number of classes. The lower the value of Gini, the purer the node.

Once trained on its sample of data, each tree classifies transactions as genuine or fraud. The classification decision of an individual tree is given by formula (7):

$$h_i(x) = class(x) \tag{7}$$

Where hi(x) is the i-th decision tree prediction.

4. Majority Voting for Final Prediction

In Random Forest, there are multiple trees that provide independent predictions. The overall classification is done by majority voting, where the class predicted by most of the trees is taken. This is illustrated as equation (8):

$$\hat{y} = mode\,\{h_1(x), h_2(x), \ldots h_N\ (x) \tag{8}$$

Where hi(x) is the prediction for each tree and N is the number of trees. If most of the trees classify a transaction as a fraud, it gets marked accordingly.

3.3 Hyperparameter Tuning for Random Forest

These are the hyperparameters that are fine-tuned for optimal model performance:

✓ Tree Count: Controls how many trees of decision exist within the model. Increasing more trees increases accuracy at the expense of computation time.

✓ Max Depth: Avoids overfitting as it cements the limit for tree depth.

✓ Minimum Samples Split: Controls how many samples can be split off a node so that splits can be made not too complicated but also unnecessary.

✓ Max Features: Tunes maximum number of features taken into consideration at every split to improve the model performance.

4 Experimental Results

4.1 Performance Evaluation Metrics

The performance of the model is assessed using Accuracy, Precision, Recall, and AUC-ROC scores. F1-score is computed using Equation (9) that gives the balance between recall and precision:

$$F1 = \frac{2x \text{ Precisionx Recall}}{\text{Precision+Recall}} \tag{9}$$

Where:

$$precision = \frac{TP}{TP+FP} \tag{10}$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{11}$$

Table 1. AUC-ROC score for fraud detection.

AUC-ROC Score	Performance Interpretation
0.90- 1.00	Excellent (Detects with high accuracy)
0.70- 0.89	Moderate (Detects with misclassifications)
0.50- 0.69	Poor (Similar to random guessing)

A Random Forest fraud detection model: As indicated in table 1, an AUC-ROC greater than 0.95 for a Random Forest, for instance, means that the model is quite successful in telling fraudulent transactions and non-fraudulent transactions apart. If the AUC is low (under 0.75) it means that model should be optimized by features like hyper parameter tuning, better feature extraction. Fig 2 Model performance representation.

Model comparison, represented as the bar chart in Fig 2, can be seen below. 2 It gives the all in one performance comparison metrics for some machine learning models like Logistic Regression, Random Forest, and XGBoost regarding the following overall metrics like Accuracy, Precision, Recall, and F1-Score. Both Random Forest and XGBoost models perform well in all the metrics so they are well suited for fraud detection.

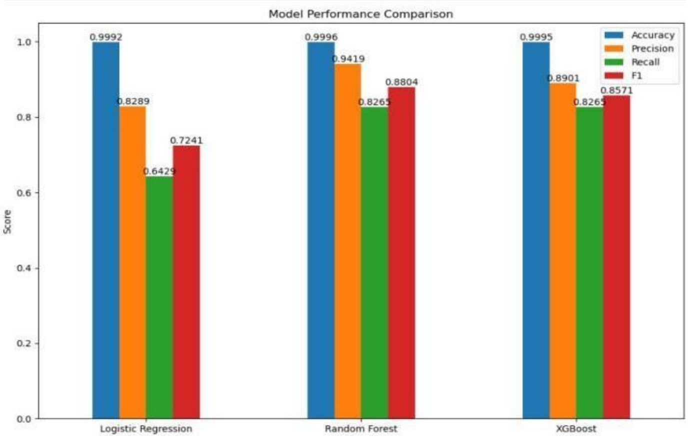

Fig. 2. Comparison analysis**.**

Although the Logistic Regression model performs well, it lacks recall, which means possible failure in the detection of fraudulent cases. The conclusions aid in the identification of best-performing model for actual use cases for fraud detection.

Table 2. Performance Metrics.

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Logistic Regression	0.93	0.82	0.64	0.72	0.93
Random Forest	0.96	0.94	0.82	0.88	0.96
Naive Bayes	0.87	0.82	0.88	0.85	0.86
XGBoost	0.95	0.89	0.82	0.85	0.92

Table 2 is a comparison study where the significant performance indicators of different models are displayed. Random Forest model boasts a highest accuracy of 96%, and precision and recall values well-balanced to achieve effective fraud detection with hardly any false positives. XGBoost also is satisfactory with accuracy of 95%. These results confirm the capability of ensemble learning techniques in fraud detection, which delivers strong and solid classification.

5 Discussion

In future, the next step to the credit card fraud detection model could be to make it more dynamic and sensitive based on how well each model and technique is accurately performing. The one crucial area of improvement is to build self-learning fraud detection systems that never stops learning and they are continuously learning new patterns of frauds as they pop-up, thereby being prepared for future threats. By finetuning to achieve better accuracy and employing the deep learning techniques, it is possible for the transformer to even better capture the complicated transaction behaviors. Furthermore, this real-time fraud detection can be extended using big data programming libraries such as Apache Spark to process large numbers of transactions in seconds. Second, there is the push for AI-driven decision making to be more transparent using some Explainable AI approach as well as ensuring that financial institutions can trust and understand fraud alerts. Finally, extending fraud detection to cross-border transactions as well as multi-bank systems will also make the world more secure and the exchange of money safer and more reliable.

6 Conclusion

In this study we compared machine learning approaches, including Random Forest, Naive Bayes and Logistic Regression, in relation to the problem of credit card fraud detection. The dataset, real transactions and formatted by PCA to preserve confidentiality, contains the relatively severe class imbalance, to solve which is mitigated through SMOTE and aims at improving the detection of fraud. The Random Forest shown to be the best performing and most stable model used. As it is commonly known, the Logistic Regression could not handle complex fraud patterns because it has linear decision bound. From those methods, Naive Bayes, was not a suitable option since it suffered from loss of accuracy because of the assumption of conditional independence that failed to hold with financial transaction data. On the other hand, Random Forest as an ensemble method worked very well in handling variance and overfitting very well and non-linearity in data well and has weal performed and have performance best among all above. It was tested by using such critical measures as precision, recall, F1-score, which simultaneously provide fraud detection and mis error reduction. The results show Random Forest has higher accuracy of classification of fraud and can be considered for practical use in financial security. There are possibilities for studies to explore hyperparameter tuning technique and explore what is the best hybrid approach in order to improve the performance of fraud detection.

References

[1] Muaz, A., Jayabalan, M., & Thiruchelvam, V. (2020). A comparison of data sampling techniques for credit card fraud detection. *International Journal of Advanced Computer Science and Applications*, *11*(6), 1–8. https://doi.org/10.14569/IJACSA.2020.0110660

[2] Alamri, M., & Ykhlef, M. (2022). Survey of Credit Card Anomaly and Fraud Detection Using Sampling Techniques. *Electronics*, *11*(23), 4003. https://doi.org/10.3390/electronics11234003

[3] Nami, S., & Shajari, M. (2018). Cost-sensitive payment card fraud detection based on dynamic Random Forest and k-NN. *Expert Systems with Applications*, *110*, 381–392. https://doi.org/10.1016/j.eswa.2018.06.011

[4] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," in *IEEE Access*, vol. 10, pp. 39700-39715, 2022, doi: 10.1109/ACCESS.2022.3166891.

[5] H. Zhu, M. Zhou, Y. Xie and A. Albeshri, "A Self-Adapting and Efficient Dandelion Algorithm and Its Application to Feature Selection for Credit Card Fraud Detection," in *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 377-390, February 2024, doi: 10.1109/JAS.2023.124008.

[6] Lucas, Y., & Jurgovsky, J. (2020). Credit card fraud detection using ML: A survey. *arXiv:2010.06479*. https://arxiv.org/abs/2010.06479

[7] E. Ileberi, Y. Sun and Z. Wang, "Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost," in *IEEE Access*, vol. 9, pp. 165286-165294, 2021, doi: 10.1109/ACCESS.2021.3134330.

[8] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. -S. Hacid and H. Zeineddine, "An Experimental Study with Imbalanced Classification Approaches for Credit Card Fraud Detection," in *IEEE Access*, vol. 7, pp. 93010-93022, 2019, doi: 10.1109/ACCESS.2019.2927266.

[9] Cherif, A., et al. (2023). Credit card fraud detection in the era of disruptive technologies: A systematic review. *Journal of King Saud University – Computer and Information Sciences, 35*(1), 145–174. https://doi.org/10.1016/j.jksuci.2022.11.008

[10] M. Zamini and G. Montazer, Credit Card Fraud Detection using Autoencoder-Based Clustering, 9th Int. Symp. Tele commun., (2019), pp. 486–491. DOI:10.1109/ISTEL.2018.8661129

[11] Carcillo, F., Le Borgne, Y.-A., Caelen, O., Bontempi, G., & others. (2018). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences, 557*, 317–331. https://doi.org/10.1016/j.ins.2019.05.042

[12] Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.-E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications, 100*, 234–245. https://doi.org/10.1016/j.eswa.2018.01.037

[13] Dang, T. K., Tran, T. C., Tuan, L. M., & Tiep, M. V. (2021). Machine Learning Based on Resampling Approaches and Deep Reinforcement Learning for Credit Card Fraud Detection Systems. *Applied Sciences*, *11*(21), 10004. https://doi.org/10.3390/app112110004

[14] T. C. Tran and T. K. Dang, Machine Learning for Prediction of Imbalanced Data: Credit Fraud Detection, Proc. IEEE 15th Int. Conf. on Ubiquitous Information Management and Communication (IMCOM), (2021), pp. 1–7. DOI:10.1109/IMCOM51814.2021.9377352

[15] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi and G. Bontempi, "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3784-3797, Aug. 2018, doi: 10.1109/TNNLS.2017.2736643.

[16] Y. Xie, M. C. Zhou, G. Liu, L. Wei, H. Zhu, and P. Meo, “A transactional-behavior-based hierarchical gated network for credit card fraud detection,” *IEEE/CAA J. Autom. Sinica*, vol. 12, no. 7, pp. 1489–1503, Jul. 2025. doi: 10.1109/JAS.2025.125243