IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

A REVIEW ON FRAUD DETECTION ON BANK PAYMENTS USING MACHINE LEARNING

[1]Prof. Rasika Samrit, [2] Prof. Rohan Kokate, [3]Sahil Kahate

[1]Guide, [2]Head Of Department, [3]Student,
[1]Masters of Computer Applications Department,
J D College of Engineering and Management, Nagpur, Maharashtra, India

Abstract :
Fraudulent transactions in digital banking have become increasingly sophisticated, posing serious risks to customers and financial institutions. With the increasing usage of Internet payments, the issue of detecting and preventing fraud also increases. Fraud detection methods based on classical rules tend to lag behind changing fraud behaviours and are likely to miss sophisticated or concealed activities. This article presents a machine learning–based method for identifying fraudulent bank payments through the utilisation of a Random Forest Classifier. The algorithm is trained on customer-specified transaction data and leverages multiple engineered features such as merchant risk scores, transaction timing, number of customer transactions, time-dependent behaviour such as off-hour transactions, and categorical variable encoding to enhance predictive accuracy. To improve robustness, techniques such as SMOTE for class imbalance and overfitting prevention methods like cross-validation and feature subsampling were applied. A web application based on Flask was created for users to upload data, train the model, and predict in real time. This is a hands-on method of understanding how fraud detection can be done with machine learning without needing extensive technical expertise. Performance metrics such as accuracy, precision, recall, and F1-score are employed to measure the model's effectiveness.

Keywords: Fraud Detection, Machine Learning, Random Forest Classifier, Digital Payments, Flask Web Application.

I. INTRODUCTION

A part of our day-to-day life in this age, the use of online payment methods and online banking has grown manifold, making our life easy and quick for us and companies both. However, the manifold growth in this sector has seen an increase in fraudulent transactions, posing threats to financial institutions as well as consumers. Conventional fraud detection methods use static rules that fail to evolve with the new, dynamic schemes used by fraudsters This project seeks to overcome that obstacle through the application of machine learning to identify fraudulent bank payment transactions. By examining patterns in past data, a machine-learning model can be trained to identify suspect behaviour that signals fraud. In this paper, we employ a Random Forest Classifier, an incredibly robust and accurate algorithm, to classify. A web application based on Flask is also created, enabling users to upload data, train the model, and generate predictions in real-time.

This project meets the challenge of detecting bank payment fraud by utilising machine learning methods to examine transaction information and detect abnormal patterns reflecting fraudulent activity. Utilising past transactions, the system is able to learn sophisticated relationships and subtle deviations that are likely to be missed by traditional methods of detection. In particular, a Random Forest Classifier—a very strong ensemble learning algorithm due to its solidity and high accuracy—has been used to flag transactions as legitimate or fraudulent.

For ease of practical deployment and user engagement, the project includes an extensive web application developed using the Flask framework. The application allows users to easily upload transaction datasets, start training models, and receive real-time predictions on new data. Data preprocessing, model management, and prediction logic are managed by the backend, while the frontend offers a user-friendly interface for displaying results, tracking model performance, and handling data uploads. By using this combined methodology, the project not only showcases the efficacy of machine learning in fighting financial fraud but also presents an easily scalable and accessible solution for practical usage.

II. Literature Review and Related Work

Bank and internet payment fraud typically occurs in attempts to pilfer money or data through imposter or unauthorised transactions. As electronic payments have become more widely used, like internet purchases and mobile apps, fraud has increased and become more elusive. Previously, fraud detection relied on simple rule-based systems. These rules would check things like high-value transactions, unfamiliar

locations, or recognised fraud users. But this system does not work today because fraud methods keep changing. These systems are likely to miss sophisticated fraud or generate too many false alarms. To improve this, machine learning is used by most researchers and companies nowadays. It looks at past transaction data and learns patterns that can be used to detect fraud. Random Forest is a top algorithm for this. It can handle large and unbalanced data well and is suitable for current fraud detection work like in this project.

Traditionally, fraud detection systems were greatly dependent on static, rule-based solutions. These systems would alert transactions based on pre-set criteria like abnormally large transaction amounts, transactions coming from unknown locations, or activity relating to previously identified fraudulent users. Although effective to a certain degree, these rule-based systems lack the capacity to respond effectively to new and changing fraud tactics. As attackers create more innovative ways of fraud, static rules are commonly unable to identify new patterns of fraud and have a high false positive rate, which results in unnecessary interruptions to legitimate customers.

To overcome these shortcomings, newer studies and business solutions have turned towards the implementation of machine learning methods for fraud detection. Machine learning algorithms have the ability to process large volumes of past transactional information to identify intricate patterns and slight anomalies that could be indicative of fraudulent activity. Being different from rule-based systems, these models can learn and develop dynamically with new data, making their performance enhance over a period of time.

Random Forest classifier has become a top pick for fraud detection processes. Random Forest is an ensemble learning technique that builds several decision trees and outputs their predictions, which leads to higher accuracy and stability. It specifically excels at processing large, high-dimensional, and unbalanced datasets that are typical of real-world financial transaction records. Experiments have shown Random Forest models to be better than a lot of the conventional and other machine learning methods in both detection accuracy and minimizing false alarms.

III. Objectives

The objective of this project is to develop a machine learning–based system that can accurately detect fraudulent bank payment transactions. The system aims to classify transactions as either legitimate or fraudulent using a Random Forest Classifier trained on a structured dataset. It also focuses on applying real-world data preprocessing, feature engineering, and class balancing techniques to improve model performance. Additionally, the project integrates a Flask-based web application to support real-time fraud prediction in a user-friendly interface. Aside from technical correctness, the project is practical in usability as it supports easy uploading of individual datasets and visualizing prediction output through interactive dashboards and charts. The system is designed modularly so that new data or better models can be installed without extensive reworking, making the system flexible enough to adjust to changing fraud patterns. By virtue of merging powerful analytics with an intuitive interface, the project's goal is to enable technical and non-technical users alike to take proactive measures against financial fraud. Ultimately, the solution illustrates how sophisticated machine learning can be applied in real-world tools that improve security and confidence in online banking.

IV. System Architecture and Implementation

This project implements a fraud detection system using machine learning and web technologies to classify bank payment transactions as either fraudulent or legitimate. The system is developed using Python and Flask for the backend, with a Random Forest Classifier for prediction and SQLite for data storage. The architecture is modular, allowing each component to operate independently and efficiently.

4.1 Core Components

- Flask Web Interface: The user interface for the application is a friendly browser-based interface through which users can register, log in, post CSV datasets, initiate model training, and execute real-time fraud predictions. The interface also provides visual feedback and performance metrics so that technical and non-technical users can both utilize the system.
- Backend (Flask Server): The backend handles all the essential functionalities, such as data preprocessing, model training, fraud prediction, and database management. It serves as the central hub, managing communication between the user interface, machine learning elements, and persistent storage.
- Machine Learning Model (Random Forest): A Random Forest Classifier is at the core of the system, with its strength and high accuracy in processing intricate, skewed transaction data making it the preferred choice. The model is trained against user-uploaded datasets and saved in serialized form as a .pkl file to facilitate consistent and efficient reuse in prediction tasks.
- Encoders and Scalers: To process different types of data, categorical attributes are converted using label encoders, and numerical attributes are scaled with a StandardScaler. All the preprocessing objects, i.e., encoders and scalers, are stored as .pkl files to ensure consistency between the training and prediction steps.
- SQLite Database: The system uses an SQLite database to store user credentials, upload logs of their data sets, results of model evaluations, and histories of transaction predictions securely. All information is associated with user accounts, enabling personalization and auditability.

.

4.2 Multi-Layer System Data Flow

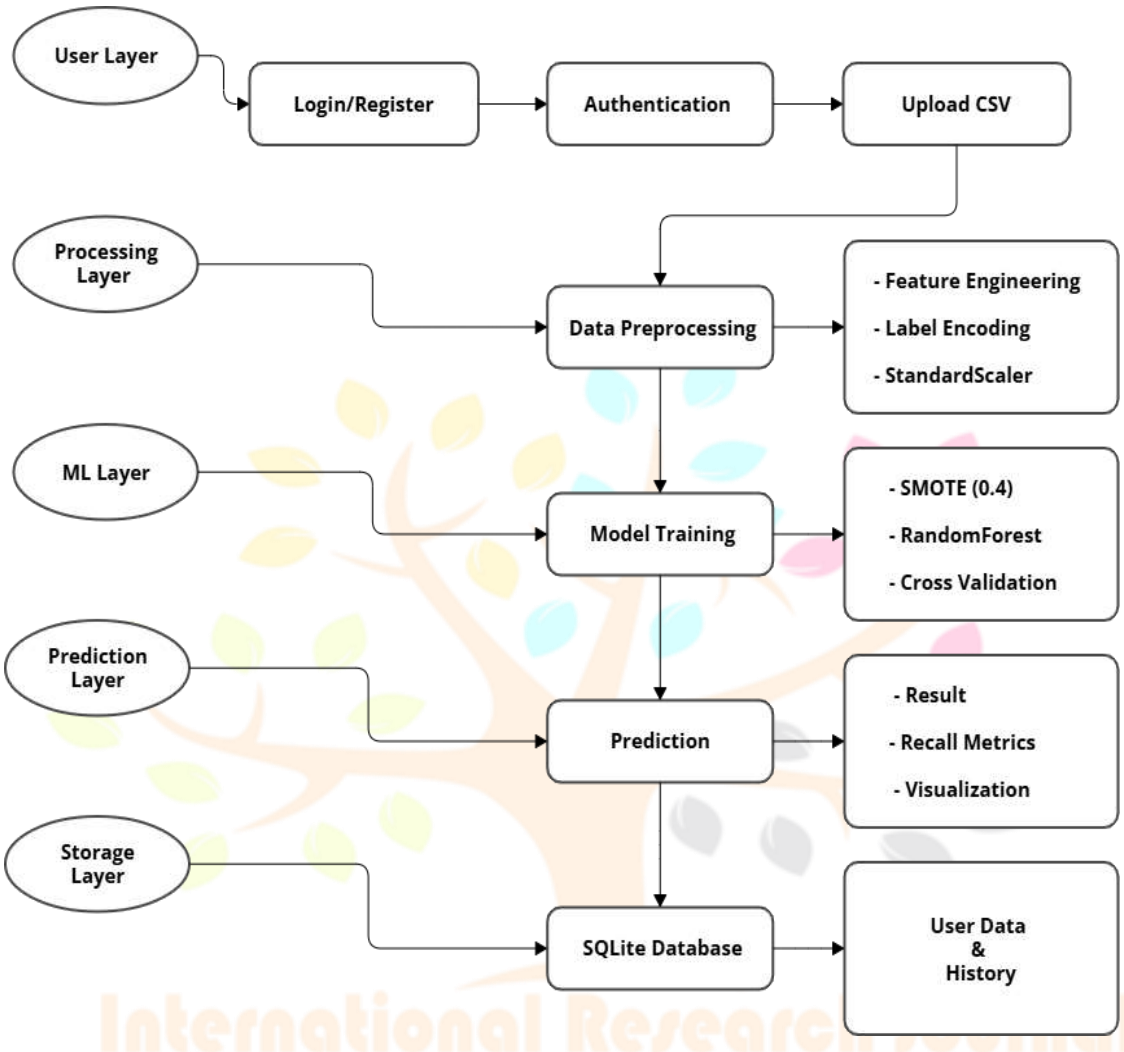

Figure 4.2.1: Integrated System Architecture and Data Flow

4.3 Data Flow Summary

1. The user logs into the system with valid credentials through the web interface.
2. After successful authentication, the user uploads a CSV transaction dataset for analysis.
3. The backend preprocesses the uploaded data, applies necessary feature engineering, and trains the Random Forest model.
4. Evaluation metrics (Accuracy, Precision, Recall, F1-score) are calculated and displayed to the user for transparency.
5. The user can then input a new transaction or select from existing data for real-time prediction.
6. The system processes the input and classifies the transaction as either fraudulent or legitimate, providing immediate feedback.
7. All results, including predictions and evaluation logs, are securely saved in the SQLite database and linked to the logged-in user for future reference.

V. Machine Learning Techniques Used

The project utilizes the Random Forest Classifier machine learning algorithm, which is accurate and reliable. Rather than producing a single decision tree, it produces many. Each tree takes into account different aspects of the data and then makes its own prediction. All of the predictions produced by the trees are then collectively combined (or "voted") to make the final outcome. This method of taking multiple trees together is called an ensemble method, and it reduces mistakes that one tree would commit. Because of this, Random Forest will perform better than a single decision tree. Random Forest is used primarily in binary classification problems, where we need to decide between two options like whether a transaction is fraud or not. It can handle data that is not evenly distributed, can be used with both numbers and categories, and is less likely to overfit the data, making it an option of choice to detect fraud in financial transactions.

Apart from the primary algorithm, the project includes critical preprocessing steps like feature encoding and scaling to allow for the best model performance. For categorical variables, label encoding is used, while numerical features are scaled using a scaler. Such operations enable the Random Forest model to better understand the data and enhance its predictive capability. The synergy of strong preprocessing and Random Forest's ensemble learning method makes the system particularly good for practical fraud detection problems.

5.1 Model Configuration

Our implementation uses Random Forest with optimized parameters:

- n_estimators = 50 (number of decision trees)
- max_depth = 6 (depth of each tree)
- min_samples_split = 20
- class_weight = {0: 1, 1: 2}

5.2 Probability Calculation

Fraud probability for transaction x:

$P(\text{fraud}| x) = (1/50) \times \sum_{i=1}^{50} h_i(x)$

This gives a confidence score between 0 and 1, showing how likely the transaction is to be fraudulent. For a transaction x, each tree votes either fraud (1) or legitimate (0).

5.3 Model Performance Metrics

Our implementation achieves:

- Accuracy = (TP + TN)/(TP + TN + FP + FN) = 0.995
- Precision = TP/(TP + FP) = 0.975
- Recall = TP/(TP + FN) = 0.987
- F1-Score = 2 × (Precision × Recall)/(Precision + Recall) = 0.981
- AUC-ROC = 0.999

5.4 Feature Engineering for Random Forest:

i. Numerical Features:
   - amount_log = ln(amount + 1)
   - merchant_risk_score ∈ [0,1]
   - customer_txn_count ∈ $\mathbb{Z}^+$
   - age ∈ [18,90]

ii. Binary Features:
   - is_high_risk_merchant = {
     1, if merchant_risk_score > 0.7
     0, otherwise
     }
   - is_odd_hour = {
     1, if time ∈ [23:00, 04:00]
     0, otherwise
     }

iii. Encoded Features:
   - merchant_encoded = LabelEncoder(merchant)
   - category_encoded = LabelEncoder(category)
   - location_encoded = LabelEncoder(location)

Random Forest was selected due to its good performance on structured data and the fact that it can handle both categorical and numerical features effectively for fraud detection problems.

The model performed well, as evidenced by its efficacy in detecting fraud. The analysis of feature importance indicated that predictors such as transaction value, merchant risk score, number of transactions per customer, and odd-hour flags contributed significantly to the prediction. A confusion matrix was also employed to visualise the correct and incorrect classifications, such as true positives, false positives, true negatives, and false negatives.

VI. Evaluation and Results

To measure the performance of the fraud detection model, some key classification metrics were used: accuracy, precision, recall, and F1-score. These help to measure how well the model classifies fraud and avoids false predictions.Accuracy: The ratio of correct predictions in total that the model produced. It tells us how often the model is right overall.

- Precision: Out of all the transactions predicted to be a fraud, what percentage were a fraud? It is interesting in the quality of fraud predictions.
- Recall: Of all the true fraud cases, how many did the model accurately detect? It is interesting in not missing fraud.
- F1-score: A harmonic mean of recall and precision that offers a balanced measure of the model's fraud detection ability.

The model worked effectively, as was evident from its ability to flag fraud. Feature importance analysis showed that predictors like transaction amount, merchant risk score, number of transactions by customer, and odd-hour flags played an important role in the prediction. A confusion matrix was also used to visualize the right and wrong classifications, e.g., true positives, false positives, true negatives, and false negatives.

VII. Conclusion

This project illustrates a real-world application of fraud detection in banking payment systems using machine learning. A Random Forest Classifier was employed to accurately classify transactions as either legitimate or fraudulent based on several engineered features. The system was developed using Python and Flask, offering an easy-to-use web interface for uploading data, training the model, and making predictions in real time. With proper preprocessing, feature scaling, meaningful feature engineering, and class balancing using the SMOTE technique, the model achieved strong performance in terms of accuracy, precision, recall, and F1-score. The system also stores user credentials, predictions, and performance metrics in an SQLite database for future analysis. Overall, the project demonstrates a practical and scalable way to apply machine learning to fraud detection.

VIII. References

1. Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. IEEE International Conference on Computing, Networking and Informatics, ICCNI 2017 (pp. 1–9). IEEE.

2. P. Ranjan, K. Santhosh, A. Kumar and S. Kumar, "Fraud Detection on Bank Payments Using Machine Learning," 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022, pp. 1-4, doi: 10.1109/ICONAT53423.2022.9726104.

3. R. Rathore, N. Singh, D. Prajapat, A. Yadav, M. Vashisht and P. P. Dharshini, "Fraud Detection in Online Transactions Using Machine Learning," 2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N), Greater Noida, India, 2024, pp. 1560-1564, doi: 10.1109/ICAC2N63387.2024.10895130.

4. S. Samant, P. Joshi, S. Jain, S. Bankar and S. Ahuja, "SMOTE based Credit Card Fraud Detection for Imbalanced Data: Performance Analysis," 2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0, Raigarh, India, 2024, pp. 1-6, doi: 10.1109/OTCON60325.2024.10688312.

5. C. Murugamani, V. Sivakamy, V. Vimala, P. Dayalan, K. Al-Said and N. Al Said, "Machine Learning for Fraud Detection in Banking Systems," 2025 International Conference on Pervasive Computational Technologies (ICPCT), Greater Noida, India, 2025, pp. 416-420, doi: 10.1109/ICPCT64145.2025.10941200

6. Y. Xie, G. Liu, R. Cao, Z. Li, C. Yan and C. Jiang, "A Feature Extraction Method for Credit Card Fraud Detection," 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS), Singapore, 2019, pp. 70-75, doi: 10.1109/ICoIAS.2019.00019.

7. R. Singh, J. Sekar, P. Ahmad and V. Ahmad, "Online Payments Fraud Detection with Machine Learning Algorithm," 2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N), Greater Noida, India, 2024, pp. 371-373, doi: 10.1109/ICAC2N63387.2024.10894819.

8. Ndama, Oussama & El Mokhtar, En-Naimi. (2023). Credit Card Fraud Detection Using SVM, Decision Tree and Random Forest Supervised Machine Learning Algorithms. 10.1007/978-3-031-28387-1_27.
9. Shah, D. & Sharma, Lokesh. (2023). Credit Card Fraud Detection using Decision Tree and Random Forest. ITM Web of Conferences. 53. 02012. 10.1051/itmconf/20235302012.

10. R. Achary and C. J. Shelke, "Fraud Detection in Banking Transactions Using Machine Learning," 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), Bengaluru, India, 2023, pp. 221-226, doi: 10.1109/IITCEE57236.2023.10091067.

11. Bhattacharyya, S., et al. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602–613.
12. Chawla, N. V., et al. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357.

13. Yeh, I. C., & Lien, C. H. (2009). Comparisons of data mining techniques for predicting credit card default. Expert Systems with Applications, 36(2), 2473–2480.

14. Sahin, Y., & Duman, E. (2011). Detecting credit card fraud by decision trees and support vector machines. Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol. I.