© May 2025 | IJIRT | Volume 11 Issue 12 | ISSN: 2349-6002

Credit Card Fraud Detection System using ML

Yuvraj Singh Pawar[1], Harsh Pawar[2], and Vilas Khedekar[3]
[1]*Student, School of Computing, MIT Art, Design and Technology University Pune, India*
[2]*M Student, School of Computing, MIT Art, Design and Technology University Pune, India*
[3]*Assistant Professor, School of Computing, MIT Art, Design and Technology University Pune, India*

***Abstract*—In this era of digital world, credit card usage has become as a preferred mode of payment for both online and offline transactions. However, this growth was followed by an alarming rise of credit card fraud, which not only poses a significant financial loss to the customer but also damages the credibility of the financial institution. Credit card fraud detection includes analyzing patterns in transaction data to flag irregular activities that may indicate fraudulent behavior. This paper explores machine learning-based approach for detecting fraudulent transactions which uses Random Forest Classifier. The system, built with a Streamlit interface, enables users to upload two datasets, visualize transaction patterns, and identify suspicious activity with high accuracy. Performance metrics including accuracy, precision, recall, and F1-score were used to assess the effectiveness of the model.**

***Index Terms*—Credit Card Fraud, Fraud Detection, Machine Learning, Random Forest Classifier**

I. INTRODUCTION

Credit card fraud has emerged as one of the most prevalent and serious forms of financial fraud in the current era of digitalization [7]. With credit cards emerging as an accepted mode of online as well as offline transactions, fraudulent activities related to them too have increased.

A credit card is a bank-issued card which allows users to make purchases on credit. Although extremely convenient, they offer an avenue for fraud. Despite them being very convenient, most credit cardholders still hold back from using their cards for regular transactions in anticipation of a potential fraud. Lack of trust here signifies a need for a secure and reliable fraud detection system [2].

Credit card fraud is being perpetrated in various manners, including card stealing, counterfeit cards, phishing, internet fraud, etc., where the cardholder is not physically present [5],[6]. These unlawful transactions not only cost the customer financially but also tarnish the images of these payment systems.

Rule-based conventional fraud systems are no longer sufficient today to keep pace with the new tactics pursued by fraudsters. Hence, data-learning-capable smart fraud detection systems are now becoming the need of the hour [3],[4].

Here, machine learning has been an apt remedy. It picks up patterns of past transactions which allow it to quite efficiently distinguish between genuine and spurious transactions [1][7]. This paper is interested in employing a Random Forest classifier to detect credit card fraud. Its objective is to develop a model which will be able to learn complex patterns and detect any suspicious activity with precision [8].

II. METHODOLOGY

This section discusses the working and deployment of the credit card fraud detection system based on the Random Forest classifier. The system is provided with two CSV files: a training dataset to train the model and an evaluation dataset for real-time fraud prediction.

1. Dataset Description

The project utilizes a training dataset is made up of historical credit card transaction data, where the transaction has been labelled as fraudulent (Class = 1) or genuine (Class = 0). The dataset contains several important features like the transaction amount, time in minutes from the first transaction, and 28 anonymized principal components (V1–V28) obtained through PCA to maintain confidentiality.

The dataset used was the Credit Card Fraud Detection dataset available on Kaggle, which contains anonymized transaction features V1–V28 resulting from PCA. The features help capture the transaction patterns maintaining user anonymity.

In addition, the system is also designed to take a secondary evaluation dataset, i.e., a fresh set of transactions entered by the user. This allows for real-time fraud detection, wherein the Random Forest model already trained was utilized in prediction and labelling potentially fraudulent transactions using patterns acquired during the training process. This two-dataset design thus allows both training of the model and efficient deployment for dynamic fraud surveillance.

2. Preprocessing

The preprocessing phase is important in a way that the data set is in clean and structured form which is a good one for training a machine learning model.

2.1. Handling Missing and Malformed Data

Data was imported from a CSV file. There are missing records in real life because of transmission faults or inconsistency in the logging process. To prevent affecting model performance or causing execution faults, data loading was told to ignore lines containing bad data (on_bad_lines='skip'). This keeps the data clean and only well-formed instances remain.

2.2. Target Variable Isolation and Feature Selection

The data has a column named "Class," which informs us whether a transaction is fraudulent (Class = 1) or not (Class = 0). This "Class" column is utilized as the target (y) for the model. The remaining columns, like Time, Amount, and 28 anonymized features (V1 to V28), are the features (X) on which the model will make its predictions. These anonymous parts were made using a procedure (PCA) to safeguard the privacy of the users while maintaining significant information regarding the transactions for the model.

2.3. Dataset Partitioning

To quantify the performance of the model numerically without risking overfitting, the dataset was divided into two sets via Scikit-learn's train_test_split function. 80% was employed in training, and 20% in testing. Random sampling is utilized to enable both subsets to have a similar proportion of fraudulent instances and legitimate instances so that they may accurately represent the distribution of data in real life, which is especially critical considering the usually unbalanced datasets found in fraud detection.

2.4. Feature Representation

All the attributes within the dataset are already in numerical format, eliminating any need for any additional preprocessing techniques.

3.Model Training with Random Forest Classifier

The fundamental machine learning algorithm used here is the Random Forest Classifier, which constructs a set of decision trees and their subsequent aggregation for prediction performance improvement. Its ability to handle high-dimensional data as well as an imbalanced class distribution makes it highly suitable for identifying credit card fraud, in which only a limited number of fraudulent transactions are included among the overall records.

3.1. Model Training

Once the data was ready, 80% of it was utilized to train a Random Forest Classifier with the help of the Scikit-learn library. Random Forest functions by creating a large number of decision trees, with each tree being trained on randomly different parts of the dataset. When prediction is to be made, every tree casts a vote, and the decision with the majority is considered the final one. This is done so that the model functions better and doesn't overfit.

3.2. Handling Missing and Malformed Data

The data was loaded from a CSV file. As is typically the case in real-world applications, incomplete entries are to be expected due to transmission errors or logging discrepancies. To avoid the entries to impact model performance or result in execution failure, the data loading process was set to skip rows with bad values (on_bad_lines='skip'). This strategy maintains data quality by keeping only clean, well-formed instances.

3.3. Performance Metrics

To ascertain the effectiveness of the model, some classification measures were computed:

•Accuracy: It finds the proportion of instances predicted correctly out of the total.

•Precision: It finds the number of the predicted transactions that were fraud and labelled as such so that there would be no false positives.

•Recall (Sensitivity): It finds the proportion of the actual fraud cases predicted correctly, which is essential so that there are no false negatives.

• F1-Score: Harmonic mean of recall and precision that provides a good balance measure best appropriate for imbalanced data sets.

• Matthews Correlation Coefficient (MCC): High-correlating measure that takes into account true positives and true negatives and false positives and false negatives, most informative in binary classification with skewed classes.

3.4. Confusion Matrix Visualization

We constructed and graphed a confusion matrix using Seaborn's heatmap. The confusion matrix is a two-dimensional table of prediction results [12]:

• True Positives – Correctly identified erroneous transactions

• True Negatives – Accurately identified actual transactions

• False Positives – Genuine transactions wrongly categorized as fraud

• False Negatives – Overlooked fraudulent transactions by the model

4. Web Based Interface using Streamlit

For reconciling the two ends of implementation and user reachability, this suggested fraud detection system is designed as an interactive web-based front end using the Streamlit platform, which is a Python package for developing reactive data-driven apps. The interface provides users either as programmers or non-programmers with the leverage of accessing and interacting with the fraud detection system [10].

Key Functionalities

Streamlit app is designed to make end-to-end user interaction including both data handling and inference during runtime possible. Its functionalities include:

• CSV Upload Feature: They are able to upload any transactional records that they may have in CSV format. The first set is utilized for model training (already pre-labelled by known results), and the second is an uploaded user-set of new data to use for real-time fraud prediction.

•Dataset Preview: The interface, upon uploading, provides a preview of the content of the two datasets in interactive tables. This option enables the user to cross-check the content and structure of the data uploaded before analysis.

• Automated Model Training and Testing: It is training the model on the training set in one input. It does internal validation in the form of an 80/20 train-test split and also calculates the important performance metrics like accuracy, precision, recall and F1-score.

• Visual Performance Feedback: Streamlit software calculates a confusion matrix and presents it in the form of a heatmap by utilizing the heatmap function within Seaborn. It is a readable graphical representation of the prediction pattern of the model.

• Real-Time Fraud Predictions: Once the model is trained, the system can apply the model directly on the second dataset that was uploaded. All the transactions are fed through it, and the fraud transactions that get identified are tagged accordingly.

III. RESULT AND ANALYSIS

This part presents the performance parameters and total performance of the Random Forest Classifier on the training set and further explains the application in real-world situations of the system.

1.Model Evaluation

After training the Random Forest Classifier on the training set, the performance of the model was assessed using a 20% holdout test set that was kept away from training. The test was conducted to observe how generalizable the model was to new data and how well it performed with respect to detecting frauds. The following key performance metrics were computed:

• The accuracy: Is the overall percentage of correctly classified transactions as fraudulent and legitimate in comparison to the number of cases. High accuracy will typically show that the model is highly accurate, though it will not capture the complete performance on unbalanced data sets.

• Precision: The number of transactions which the model has detected as fraudulent and really are fraudulent. A high precision indicates the ability of the model not to raise false alarms (false positives), which in fraud detection is critical to avoid unnecessary investigation.

• Recall (or sensitivity or true positive rate): Indicates the ratio of actual fraud transactions that were properly flagged by the model. Good recall is crucial in keeping the number of fraud transactions that are unknown low.

• F1 Score: Harmonic mean between recall and precision, giving an equally weighted measure considering both false positives and false negatives. It is particularly useful in imbalanced datasets cases such as fraud detection where the instances of fraud are much fewer than valid ones.

2.Confusion Matrix Analysis

To further improve the comprehension of the model's performance, a confusion matrix heatmap is presented in the application. The visual graph allows users to see at a glance how well the model is separating fraudulent and genuine transactions. The confusion matrix consists of the following main elements [12]:

•True Positives (TP): They are those fraudulent transactions identified as fraudulent by the model correctly. A greater number of true positives means the model is actually catching fraud correctly.

•True Negatives (TN): These are the authentic transactions that have been properly categorized as non-fraudulent. High true negatives indicate that the model is not flagging valid transactions as fraudulent wrongly.

•False Positives (FP): These are valid transactions that the model incorrectly identifies as fraudulent. A large proportion of false positives may result in unnecessary investigations or customer frustration, so it is important to keep this category at a minimum.

• False Negatives (FN): These are false transactions which the model does not detect, marking them as regular. This is an important measure in fraud detection systems since failing to detect false transactions can result in financial loss and security breaches.

3.Fraud Detection on Evaluation Data

After the Random Forest Classifier model is trained using the training dataset, it can be used to make real-time fraud predictions on an evaluation dataset. This involves the following steps:

•Feature Alignment: The system aligns the features of the untraining input dataset with those of the training dataset. This means that all columns are aligned correctly and have the same structure and format.

•Prediction: The model subsequently predicts the class for every transaction in the untraining dataset. This is achieved by sending the feature set through the trained classifier, which outputs a predicted class label (legitimate = 0 or fraudulent = 1) to every transaction.

•Flagging Fraudulent Transactions: Transactions that are predicted as fraudulent (Predicted_Class = 1) are marked. These records are suspicious and potentially dangerous, and they require further investigation or intervention.

•Predicted Data Display: The system then shows a table of all transactions flagged as fraudulent. The table gives users a detailed representation of the predicted fraudulent transactions, including the associated feature values and predicted class labels.

• CSV Export: To export the detected data to analyse or report further, the system provides a download button so users can export the flagged data in CSV format. This enables users to extract the detected fraudulent transactions and incorporate them into their workflow or database for further action, e.g., investigation or notification of users.

IV. CONCLUSION

In this project, we created a web-based credit card fraud detection system using a Random Forest Classifier. The system was intended to solve real-life problems in fraud detection by learning the model from a train set of training credit card transactions. The train dataset includes fraudulent (Class = 1) and genuine (Class = 0) transactions that enable the model to learn discriminatory patterns against fraud. The model was tested for its performance on an evaluation dataset comprised of unseen data to test the ability of the model to generalize and predict fraud under real-world situations.

Through training the model using the training data, the Random Forest Classifier had great performance metrics in terms of high accuracy, precision, recall, and F1-score that indicates how well it performs in detecting fraudulent transactions without allowing false positives. The system was created using a friendly user interface through Streamlit such that the users will be able to interact with the model efficiently, upload data sets, view essential performance metrics, and receive real-time fraud predictions. The software not only allows training the model on a training data set but also allows users to use the trained model for making predictions on an evaluation data set, where predictions are being made on unseen transactions.

The Random Forest model proved to work well with imbalanced datasets, a prevalence problem in fraud detection cases where there are few fraudulent transactions versus legitimate ones. The system also needed little hyperparameter adjustment, which was an excellent solution. The integration with Streamlit even made it more user-friendly, allowing users of different backgrounds, including those outside of data science stakeholders, to use the system, view results, and understand predictions easily.

This solution is centred on the efficiency and simplicity of employing traditional machine learning methods, i.e., Random Forests, for constructing an efficient and quick fraud detection system. By using careful feature engineering, interactive visualization, and usability-focused deployment methods, the system proves that real-time fraud detection is feasible even with relatively simple models in a readily usable experience.

V. FUTURE WORK

While the present Streamlit-based credit card fraud prediction system is interactive, as well as operational for fraud prediction, there are a number of areas where further research can help improve its performance, functionality, and use in real-life scenarios. They are:

1. Real-Time Fraud Detection Integration

The model can now provide a batch prediction for an evaluation set. For the application to be more beneficial to use in actual applications, there may be some future work that involves real-time fraud detection. This will allow users to see and determine fraudulent transactions in real-time while being executed rather than model retraining in batches.

2 Improved User Interface and Usability

While the Streamlit interface itself is intuitive and easy to manage, it can still be made easier. The future versions can include features such as data discovery feature (e.g., interactive transaction rate charts), improved performance measurement, and an even more flat appearance to drive through the fraud detection for the users. The non-technical users can be guided through a tutorial or introduction period for acclimatization with the system.

3.pettoizable Uploading of dataset

The users at this time can upload a training data set and a test data set in CSV file format. In future versions, the system will be able to accept other file formats (Excel, JSON, etc.) and pre-cleaning or pre-processing data before training of the model will be achievable. It might provide options to handle missing values, feature selection, or normalization of features standalone, which would provide the user with greater flexibility.

4. Mode explainability

For explainability and credibility of predictions, certain of the future work may incorporate explainability features. Adding SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) will make the users aware of why a specific fraud prediction has been obtained, allowing the financial institutions to explain why they are doing it and making decisions on that basis. This will also increase the trust of the users in the model predictions.

5. Performance with Imbalanced Data:

The system currently employs a Random Forest classifier, which is not affected by imbalanced datasets, but the imbalance between real and simulated transactions persists. Subsequent releases of the system can incorporate more sophisticated techniques like SMOTE (Synthetic Minority Over-sampling Technique) or ADASYN (Adaptive Synthetic Sampling) to balance the training dataset, hopefully obtaining maximum recall and minimum false negatives.

6. Model Retraining and Continuous Learning:

To enhance the system in identifying fraud since fraud scams evolve, the app can be rolled out with continuous model learning and retraining. With model auto-retraining appended to new transactional data, the system would be up to date with newer future fraud schemes, with the long-term detection rate improved.

7. Scalability for Big Data

The current system provides reasonable performance for small sets but, when dealing with gigantic applications of the magnitude of millions of transactions, there is some performance compromise. Optimization to process big data can be future research and may include cloud technology-based distributed computing in order to make the model scale enterprise-class usage.

8. Better Visualization of Model Performance

Although the current application is able to create and print a confusion matrix, other types of visualizations can be embedded so that the users can have a better view of the performance of the model. Precision-Recall curves, ROC curves, or feature importance plots are some types of visualizations which can enable the users to interpret the performance of the model better and the weaknesses of the model, and use them in informing decision-making when carrying out a fraud investigation.

9. Data Privacy and Security Improvements:

The fact that the system processes sensitive monetary information means that data privacy and security are of paramount importance. Upcoming releases can include data encryption upload, secure authentication process utilization, and compliance with data protection laws (i.e., GDPR). Additionally, anonymizing sensitive customer information without compromising fraud prediction performance would be an important improvement.

10. Multi-Language and Multi-Currency Transaction Support

Since credit card fraud detection is a global issue, its use to support multi-language interface and multi-currency payment would further increase its usage. It would allow banks in countries and regions to utilize the system to their maximum benefit, according to local specifications and local payment habits.

11. Post-Prediction Actions

Such a feature beneficial to incorporate into the system is to have a feature to take action upon fraud identification. For example, a feature for alerting administrators or users with automated notifications whenever there is system identification of fraud. Another beneficial feature is to incorporate a feature to flag or quarantine potential transactions for inspection, which can be beneficial for banks.

VI. ACKNOWLEDGMENT

The authors would like to thank the faculty and staff of the Department of Computer Science at MIT ADT University for their guidance and support throughout this research. Special thanks are extended to our faculty guide Prof. Vilas Khedkar for his invaluable feedback and mentorship.

The authors also acknowledge the use of open-source machine learning and data analysis libraries such as scikit-learn and TensorFlow, which played a critical role in the development and optimization of the AI-powered Credit Card Fraud Detection System. Their contributions enabled the implementation of effective algorithms and real-time fraud detection models.

REFERENCES

[1] T A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating Probability with Undersampling for Unbalanced Classification," in Proc. 2015 IEEE Symposium Series on Computational Intelligence (SSCI), Cape Town, South Africa, Dec. 2015. DOI: 10.1109/SSCI.2015.33.

[2] S. B. E. Raj and A. A. Portia, "Analysis on Credit Card Fraud Detection Methods," in Proc. IEEE.

[3] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit Card Fraud Detection Using Machine Learning Techniques: A Comparative Analysis," in Proc. IEEE.

[4] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," Expert Systems with Applications, vol. 40, no. 15, pp. 5916-5923, 2013.

[5] A. Shen, R. Tong, and Y. Deng, "Application of classification models on credit card fraud detection," June 2007.

[6] Y. Kou, C-T. Lu, S. Sinvongwattana, and Y-P. Huang, "Survey of fraud detection techniques," in Proc. 2004 IEEE International Conference on Networking Sensing & Control, 2004.

[7] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using Bayesian and neural networks," in Proc. 1st International Naiso Congress on Neuro-Fuzzy Technologies, pp. 261-270, 2002.

[8] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," Decision Support Systems, vol. 50, no. 3, pp. 602-613, 2011.

[9] S. Patil, H. Somavanshi, J. Gaikwad, A. Deshmane, and R. Badgujar, "Credit card fraud detection using decision tree induction algorithm," International Journal of Computer Science and Mobile Computing (IJCSMC), vol. 4, no. 4, pp. 92-95, 2015, ISSN 2320-088X.

[10] Amlan Kundu, Suvasini Panigrahi, Shamik Sural, and Arun K. Majumdar, "BLAST-SSAHA hybridization for credit card fraud detection," IEEE Transactions on Dependable and Secure Computing, vol. 6, no. 4, pp. 309-315, Oct.-Dec. 2009.

[11] B. M. Thippeswamy, H. V. Ramachandra, R. S., R. Salam, and M. Pai, "TextVerse: A Streamlit web application for advanced analysis of PDF and image files with and without language models," IEEE, [Online]

[12] Kaiyong Deng, Ru Zhang, Hong Guo, Dongfang Zhang, Wenfeng Jiang, and Xinxin Niu,

"Analysis and study on detection of credit fraud in e-commerce," 2011 IEEE.

[13] Swaminathan, S., & Tantri, B. R. (2024, November). Confusion matrix-based performance evaluation metrics. African Journal of Biomedical Research, 27(4S), 4023–4031.