Credit Card Fraud Detection

Mrs. D.Mamatha[1], M.Hemanth [2], K.Srikanth[3], N.Prakash [4], S.Bharath [5]
[1]*Assistant Professor, Department of CSE (Cyber Security), Sphoorthy Engineering College, Hyderabad, Telangana, India*
[2,3,4,5] *Members, Department of CSE (Cyber Security), Sphoorthy Engineering College, Hyderabad, Telangana, India*

Abstract— This study presents a machine-learning-based system designed for the real-time identification of credit card fraud in digital transactions. As online payments continue to grow rapidly, the demand for fast and reliable fraud detection has become critical. A primary challenge in this domain is the severe class imbalance present in transaction data, where legitimate records vastly outnumber fraudulent ones.

The framework incorporates comprehensive data preprocessing, targeted feature engineering, and class balancing through the Synthetic Minority Oversampling Technique (SMOTE). Two supervised learning algorithms—Decision Tree and Random Forest—were implemented and thoroughly assessed using standard evaluation metrics. Results show that the Random Forest classifier achieves superior performance across precision, recall, F1-score, and overall accuracy, while also minimizing false positives. The proposed approach is efficient, scalable, and ready for integration into banking and financial platforms to enhance security and reduce losses.

I. INTRODUCTION

The widespread adoption of digital payment systems has transformed how people conduct financial transactions, delivering greater speed and convenience. However, this shift has also opened new avenues for fraud, especially in credit card operations. Banks and financial institutions face mounting pressure to detect suspicious activities amid exploding transaction volumes and increasingly sophisticated fraud methods.

Successful fraud detection systems must identify anomalies quickly while keeping false alarms to a minimum. The inherent class imbalance—fraudulent cases often representing less than 0.2 % of all records—makes the problem particularly difficult. Conventional rule-based systems struggle to adapt to emerging fraud patterns. Machine learning offers a more dynamic solution by learning complex relationships directly from historical transaction data. This research develops an effective fraud detection pipeline that combines careful data preparation, SMOTE-based balancing, and rigorous model evaluation to deliver high accuracy, robustness, and practical applicability in real-world banking environments.

II. LITERATURE SURVEY

Credit card fraud detection has been explored through various computational approaches over the years. Early methods relied on fixed rule-based systems that flagged transactions based on predefined thresholds or known suspicious behaviours. Although computationally lightweight, these approaches lacked the flexibility to handle novel fraud tactics.

The transition to machine learning introduced supervised techniques such as Logistic Regression and Decision Trees, which learn decision boundaries from labeled data. While these models improved detection rates, they frequently performed poorly on highly imbalanced datasets. Ensemble methods, particularly Random Forest and Gradient Boosting, later emerged as stronger alternatives due to their ability to handle high-dimensional features, resist overfitting, and deliver robust predictions.

Unsupervised approaches, including clustering and anomaly detection algorithms, have also been investigated for scenarios where labeled fraud data is scarce. However, these methods often generate higher false-positive rates. Recent research emphasizes the

importance of thorough preprocessing, feature selection, and resampling strategies like SMOTE to mitigate class imbalance. Despite these advancements, issues related to real-time processing and adaptation to evolving fraud strategies remain active areas of investigation.

III. EXISTING SYSTEM

Current fraud detection systems primarily depend on rule-based and statistical methods that flag transactions based on predefined rules (e.g., unusual amounts, frequency, or location). Although computationally lightweight, these systems require constant manual updates and cannot easily adapt to emerging fraud patterns.

Some implementations incorporate basic machine learning algorithms such as Logistic Regression and Decision Trees. These offer improved performance over pure rule-based systems but still struggle with severe class imbalance. Unsupervised approaches like clustering and anomaly detection help detect deviations from normal behavior; however, they frequently generate excessive false alarms, limiting their practical reliability.

In summary, existing solutions suffer from limited adaptability, moderate accuracy, and poor handling of data imbalance, underscoring the demand for more advanced, scalable alternatives.

IV. SYSTEM ARCHITECTURE

The proposed credit card fraud detection system follows a modular, end-to-end pipeline designed for efficient processing of transactional data. It begins with data acquisition, ingesting datasets that include features such as transaction time, amount, and anonymized variables (V1–V28), along with the binary class label (fraudulent or legitimate).

Data then enters a preprocessing stage to ensure quality and consistency. Irrelevant or redundant features (e.g., raw time and amount) are removed or transformed, and standardization is applied to normalize numerical values. Missing values and inconsistencies are handled to maintain data integrity.

Next, feature engineering and correlation analysis are performed to identify the most predictive attributes and eliminate redundancy. To counter the severe class imbalance, SMOTE is applied to synthesize additional minority-class samples, enabling balanced training and better model learning.

The balanced dataset feeds into the modeling phase, where Decision Tree and Random Forest classifiers are trained. Random Forest is chosen as the primary model due to its high accuracy, resistance to overfitting, and strong performance on high-dimensional data. Models are trained on historical patterns to distinguish fraudulent from genuine transactions.

Model performance is thoroughly evaluated using accuracy, precision, recall, F1-score, and confusion matrices. The best-performing model is selected for deployment. In the final stage, the system classifies incoming transactions in real time, generating alerts or blocking suspicious activity. The architecture is scalable, reliable, and ready for integration with financial platforms.

V. SYSTEM DESIGN

Credit card fraud detection typically involves several modules or components that work together to identify and prevent fraudulent activities.

1. Data Collection and Preprocessing.
2. Feature Engineering.
3. Machine Learning Modules.
4. Real-Time Monitoring and Detection
5. Model Evaluation and Refinement
6. Integration and Deployment.

Random Forest Algorithm

Random Forest is an ensemble learning algorithm that constructs multiple decision trees during training and combines their outputs to produce a final prediction. It excels in classification and regression tasks, offering high accuracy, resilience to noisy data, and the ability to work effectively even when features are unscaled or contain missing values. The algorithm operates as follows:

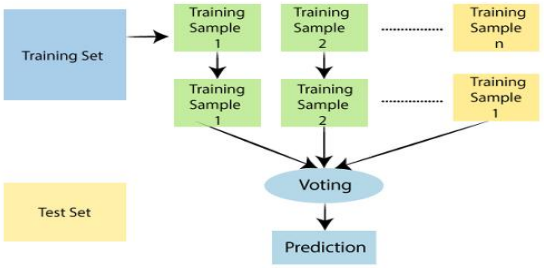
1. Random subsets of the training data are selected with replacement (bootstrap sampling).

2. A decision tree is built for each subset using a random selection of features at each split.

3. This process is repeated to generate a large collection ("forest") of trees.

4. For a new transaction, every tree provides a prediction; the final class label is determined by majority voting (classification) or averaging (regression).

By averaging the predictions of many diverse trees, Random Forest significantly reduces variance and overfitting compared to a single decision tree, resulting in more stable and reliable performance.



Random Forest is a supervised machine learning algorithm that belongs to a technique called ensemble learning. In simple terms, ensemble learning means combining multiple models to build a stronger and more accurate prediction system. In the case of Random Forest, it uses many decision trees of the same type and combines their results to make better predictions. Because it creates a collection (or "forest") of decision trees, it is called a Random Forest. This algorithm is versatile and can be used for both classification and regression tasks.

VI. IMPLEMENTATION AND RESULTS

Exploratory Data Analysis (EDA)
The dataset was loaded and examined using Python libraries (pandas, matplotlib, seaborn). No missing values were present. The "Amount" feature was standardized to bring it onto a comparable scale with other variables, and the "Time" column was removed as it carried no useful predictive information. A bar chart of the target class distribution confirmed extreme imbalance: legitimate transactions accounted for more than 99 % of the records. To address this, SMOTE was applied after splitting the data into training and test sets, generating synthetic fraudulent samples to balance the classes for model training.

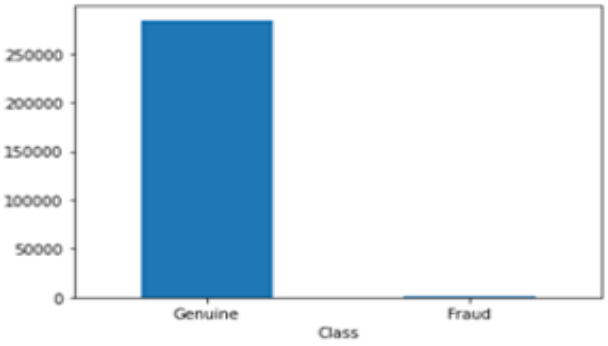


Figure 1: Complete the Exploratory Data Analysis on the dataset.

We can observe from the above bar graph that the genuine transactions are over 99%. So, to avoid this problem we can apply the scaling techniques on the "Amount" feature to transform them to the range of values. We will remove the "Amount" column and add a new column with the scaled values in its place. We will also remove the "Time" column as it is not required.

Model Training and Evaluation

Decision Tree and Random Forest classifiers were trained on the balanced training set. Predictions on the held-out test set were evaluated using accuracy, precision, recall, and F1-score, along with confusion matrices.

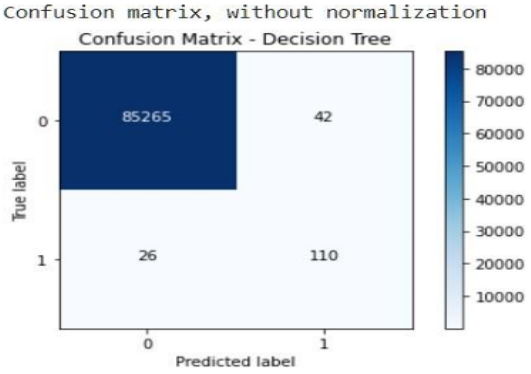


Figure 2: Train and Evaluate the Models

Decision Tree Performance

- Accuracy: 0.99920
- Precision: 0.72368
- Recall: 0.80882
- F1-score: 0.76389

Now if I visualize the confusion matrix of, the Random Forest model

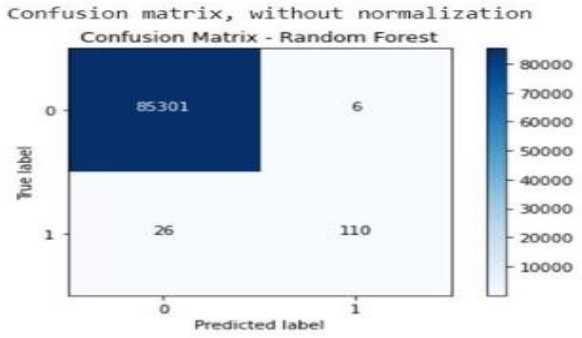


Figure 3: Confusion Matrix Without Normalization

Random Forest Performance

- Accuracy: 0.99963
- Precision: 0.94828
- Recall: 0.80882
- F1-score: 0.87302

The Random Forest model demonstrated clear superiority, delivering higher precision and a better overall balance between detecting fraud and limiting false alarms.

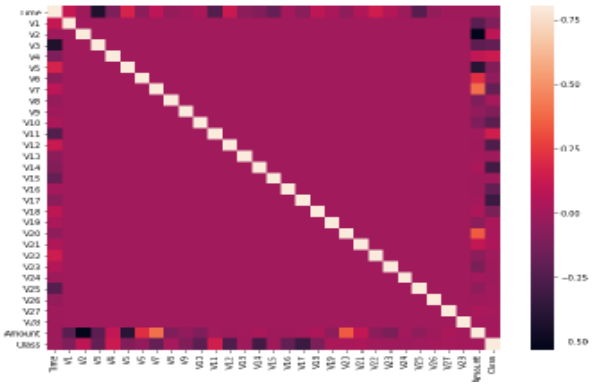
Data Preparation

The below figure shows the structure of the dataset where all attributes are shown, with their type, in addition to glimpse of the variables within each attribute, as shown at the end of the figure the Class type is integer which I needed to change to factor and identify the 0 as Not Fraud and the 1 as Fraud to ease the process of creating the model and obtain visualizations.

```
'data.frame':   284807 obs. of  31 variables:
 $ Time   : num  0 0 1 1 2 2 4 7 7 9 ...
 $ V1     : num  -1.36 1.192 -1.358 -0.966 -1.158 ...
 $ V2     : num  -0.0728 0.2662 -1.3402 -0.1852 0.8777 ...
 $ V3     : num  2.536 0.166 1.773 1.793 1.549 ...
 $ V4     : num  1.378 0.448 0.38 -0.863 0.403 ...
 $ V5     : num  -0.3383 0.06 -0.5032 -0.0103 -0.4072 ...
 $ V6     : num  0.4624 -0.0824 1.8005 1.2472 0.0959 ...
 $ V7     : num  0.2396 -0.0788 0.7915 0.2376 0.5929 ...
 $ V8     : num  0.0987 0.0851 0.2477 0.3774 -0.2705 ...
 $ V9     : num  0.364 -0.255 -1.515 -1.387 0.818 ...
 $ V10    : num  0.0908 -0.167 0.2076 -0.055 0.7531 ...
 $ V11    : num  -0.552 1.613 0.625 -0.226 -0.823 ...
 $ V12    : num  -0.6178 1.0652 0.0661 0.1782 0.5382 ...
 $ V13    : num  -0.991 0.489 0.717 0.508 1.346 ...
 $ V14    : num  -0.311 -0.144 -0.166 -0.288 -1.12 ...
 $ V15    : num  1.468 0.636 2.346 -0.631 0.175 ...
 $ V16    : num  -0.47 0.464 -2.89 -1.06 -0.451 ...
 $ V17    : num  0.208 -0.115 1.11 -0.684 -0.237 ...
 $ V18    : num  0.0258 -0.1834 -0.1214 1.9658 -0.0382 ...
 $ V19    : num  0.404 -0.146 -2.262 -1.233 0.803 ...
 $ V20    : num  0.2514 -0.0691 0.525 -0.208 0.4085 ...
 $ V21    : num  -0.01831 -0.22578 0.248 -0.1083 -0.00943 ...
 $ V22    : num  0.27784 -0.63867 0.77168 0.00527 0.79828 ...
 $ V23    : num  -0.11 0.101 0.909 -0.19 -0.137 ...
 $ V24    : num  0.0669 -0.3398 -0.6893 -1.1756 0.1413 ...
 $ V25    : num  0.129 0.167 -0.328 0.647 -0.206 ...
 $ V26    : num  -0.189 0.126 -0.139 -0.222 0.502 ...
 $ V27    : num  0.13356 -0.00898 -0.05535 0.06272 0.21942 ...
 $ V28    : num  -0.0211 0.0147 -0.0598 0.0615 0.2152 ...
 $ Amount: num  149.62 2.69 378.66 123.5 69.99 ...
 $ Class  : int  0 0 0 0 0 0 0 0 0 0 ...
```

Correlation between attributes “Image from R”

The correlations between all the of the attributes within the dataset are presented in the figure below.



VII. CONCLUSION

This research developed a practical machine-learning framework for credit card fraud detection that effectively tackles data preprocessing, feature engineering, and class imbalance through SMOTE. By balancing the dataset, the models were able to learn meaningful patterns from the rare fraudulent cases.

Among the tested classifiers, Random Forest consistently outperformed the Decision Tree, achieving the highest scores across key metrics while maintaining strong recall and low false-positive rates. The system is computationally efficient and scalable,

making it well-suited for real-time deployment in financial institutions to minimize fraud-related losses and improve customer trust.

Future work could explore deep learning architectures, online streaming pipelines, and adaptive retraining mechanisms to handle evolving fraud patterns more dynamically.

REFERENCES

[1] Adepoju, O., Wosowei, J., lawte, S., & Jaiman, H. (2019). Comparative evaluation of credit card fraud detection using machine learning techniques. 2019 Global Conference for Advancement in Technology (GCAT). https://doi.org/10.1109/gcat47503.2019.8978372

[2] Alenzi, H. Z., & Aljehane, N. O. (2020). Fraud detection in credit cards using logistic regression. International Journal of Advanced Computer Science and Applications, 11(12). https://doi.org/10.14569/ijacsa.2020.0111265

[3] Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017). Credit card fraud detection using Machine Learning Techniques: A Comparative Analysis. 2017 International Conference on Computing Networking and Informatics (ICCNI). https://doi.org/10.1109/iccni.2017.8123782

[4] Bhanusri, A., Valli, K. R. S., Jyothi, P., Sai, G. V., & Rohith, R. (2020). Credit card fraud detection using Machine learning algorithms. Journal of Research in Humanities and Social Science, 8(2), 04-11.

[5] Credit card statistics. Shift Credit Card Processing. (2021, August 30). Retrieved from https://shiftprocessing.com/credit-card.

[6] Daly, L. (2021, October 27). Identity theft and credit card fraud statistics for 2021: The ascent. The Motley Fool. Retrieved from https://www.fool.com/the-ascent/research/identity-theft- credit-card-fraud-statistics.

[7] Dheepa, V., & Dhanapal, R. (2012). Behavior based credit card fraud detection using support vector machines. ICTACT Journal on Soft Computing, 02(04), 391–397. https://doi.org/10.21917/ijsc.2012.0061