Credit Card Fraud Detection Using Random Forests with Enhanced Recall for Imbalanced Datasets

Project

by

Nwala Al-Bright Okechukwu

Department of Computer Science

College of Science Evangel University

Akeze, Ebonyi State

Abstract

Credit card fraud is a significant challenge for financial institutions and consumers, leading to billions of dollars in losses annually. This project develops a machine learning-based system to detect fraudulent transactions using the Random Forest classifier. The system addresses class imbalance, a common issue in fraud detection, by employing class weights to improve the recall of fraudulent transactions. Exploratory data analysis (EDA) reveals important transaction patterns, feature correlations, and fraud distribution. The Random Forest model achieves high performance, with an accuracy of 99.95% and a recall of 77% for fraudulent transactions, balancing precision and recall effectively. This project contributes a scalable and interpretable fraud detection solution that can be deployed in real-world scenarios to minimize financial risks.

Keywords: Credit Card Fraud, Machine Learning, Random Forest, Class Imbalance, Fraud Detection, Precision, Recall, Feature Importance, Exploratory Data Analysis (EDA), Financial Systems

Introduction

Credit card fraud remains a persistent issue in the financial sector, exacerbated by the rapid growth of online transactions and digital payment systems. Fraudulent activities not only lead to financial losses but also erode consumer trust and require significant operational resources to address.

Traditional rule-based fraud detection systems are limited by their inability to adapt to evolving fraud techniques. Machine learning offers a dynamic alternative by leveraging patterns and anomalies in transaction data to identify fraudulent behavior. However, several challenges persist:

1. **Class Imbalance**: Fraudulent transactions represent a small fraction of total transactions, making it difficult for models to detect them effectively.
2. **Model Interpretability**: Financial institutions require interpretable models to meet regulatory and stakeholder demands.
3. **Scalability**: Models must process vast amounts of data in real-time for effective fraud prevention.

This project aims to address these challenges by:

- Using Random Forests, a supervised machine learning algorithm, to classify transactions as normal or fraudulent.
- Improving recall for fraudulent transactions through class-weight adjustments.
- Extracting actionable insights into transaction behavior using exploratory data analysis (EDA).

Literature Review

Several approaches have been proposed for fraud detection, each with its strengths and weaknesses:

1. Logistic Regression

   Bhattacharyya et al. (2011) demonstrated the use of logistic regression for fraud detection, emphasizing its simplicity and computational efficiency. However, the algorithm struggles with imbalanced datasets and fails to model nonlinear relationships effectively.

   Drawback: Inability to handle complex, high-dimensional data.

   Improvement: Random Forests can model nonlinear relationships and naturally handle feature interactions.

2. Support Vector Machines (SVM)

Chawla et al. (2002) explored the use of SVMs for fraud detection. While robust, SVMs require significant computational resources for large datasets and are difficult to interpret.

Drawback: Scalability issues and lack of interpretability.

Improvement: Random Forests are computationally efficient and provide interpretable feature importance metrics.

3. **Neural Networks**

   He and Garcia (2009) highlighted the power of neural networks in modeling complex patterns in fraud detection. However, their black-box nature makes them unsuitable for regulated industries like finance.

   Drawback: Lack of transparency and high computational costs.

   Improvement: Random Forests offer interpretability while maintaining competitive accuracy.

4. **Imbalanced Learning**

   Techniques like SMOTE (Synthetic Minority Oversampling Technique) and undersampling have been widely used to address class imbalance (Chawla et al., 2002). However, these methods can introduce noise or reduce the training dataset size.

   Drawback: Potential overfitting or data loss.

   Improvement: This project leverages class-weighted Random Forests to handle imbalances directly without modifying the dataset.

Methodology

System Architecture

The fraud detection system is designed with a modular architecture, comprising the following components:

1. **Data Input**: Transaction data from financial systems.
2. **Data Preprocessing**: Cleaning, normalization, and feature selection.
3. **Feature Selection**: Identifying important features that contribute to fraud detection.
4. **Model Training:** Using Random Forests to train a robust classifier.
5. **Fraud Detection**: Classifying transactions as normal or fraudulent.

6. **Reporting**: Providing actionable insights and performance metrics.

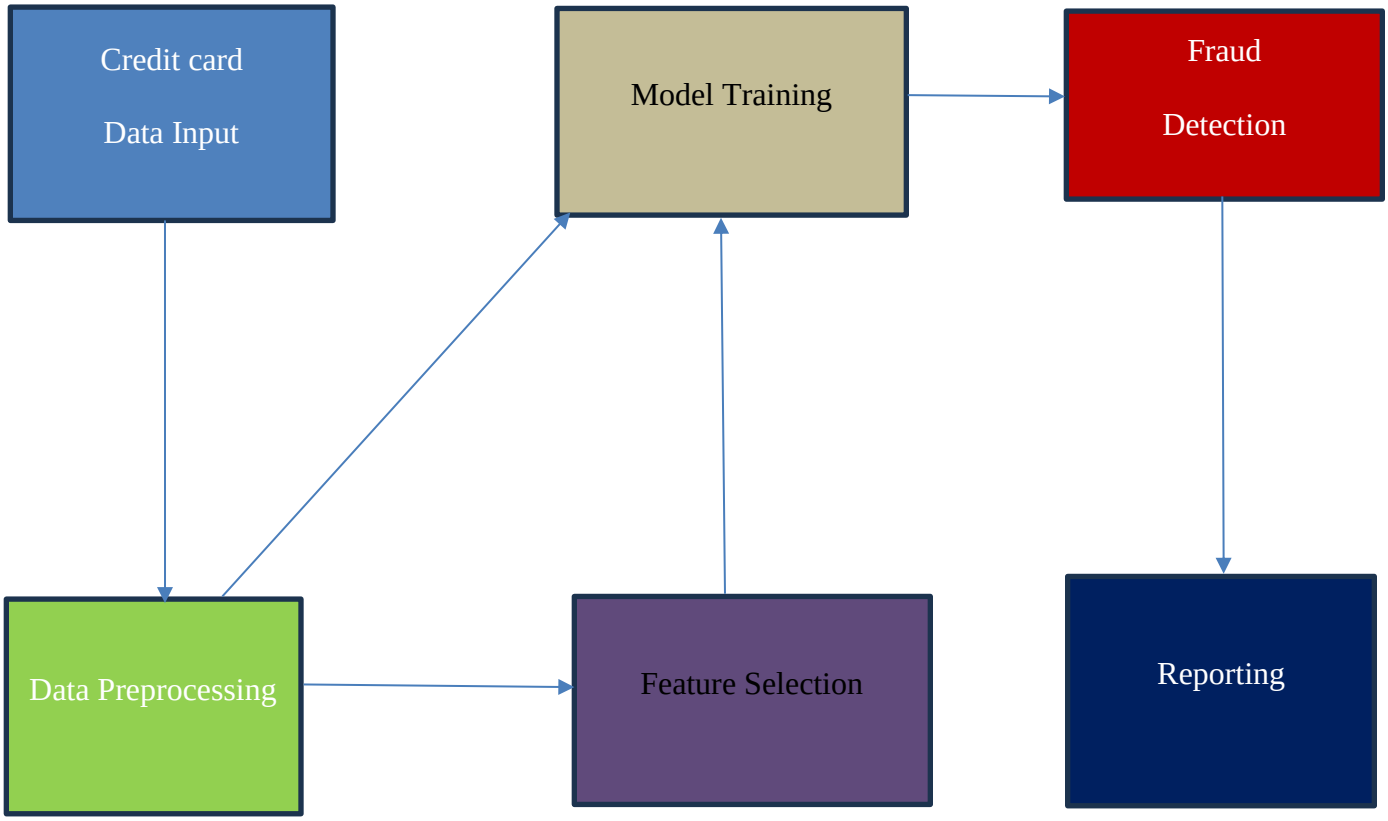


Figure 1: System Design Architecture

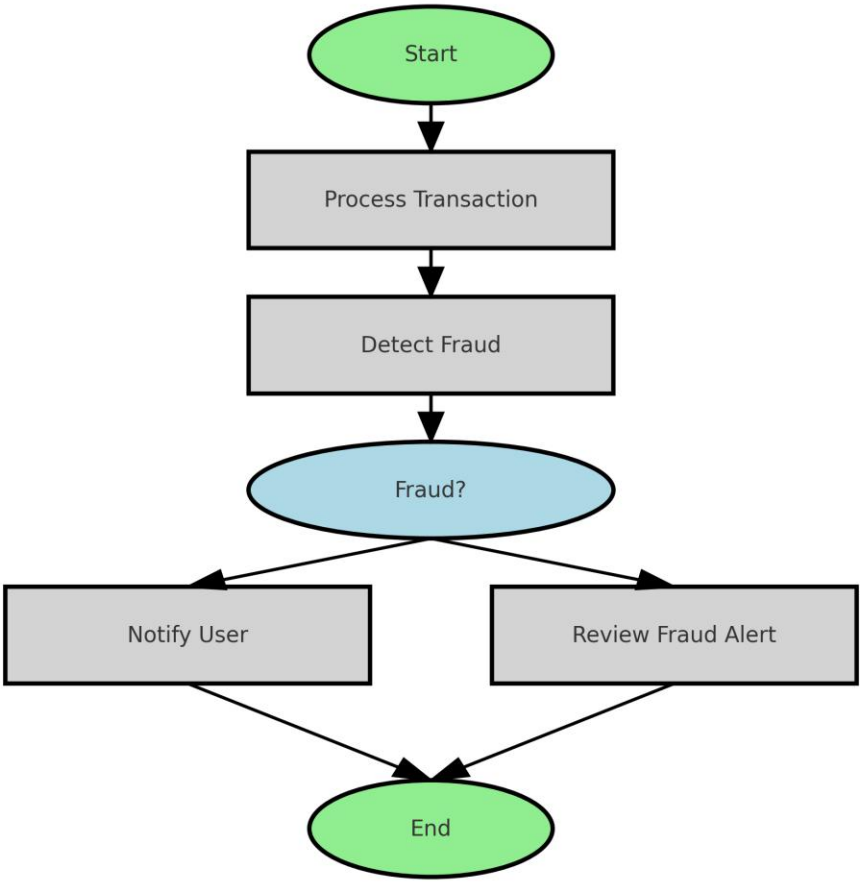


Figure 2: Flow Chart

The **Flowchart** delves deeper into the operational flow of the fraud detection process, outlining how the system handles transactions step by step. It begins with the **Start** node, symbolizing the initiation of a financial transaction by a user. This step represents the trigger for the entire fraud detection workflow, setting the system in motion.

The next step, **Process Transaction**, involves the system receiving and recording the transaction details. At this stage, data such as the transaction amount, location, time, and account information are logged. This data is essential for the subsequent analysis and serves as the foundation for fraud detection.

Once the transaction data is processed, the system moves to the **Detect Fraud** stage. Here, the transaction is analyzed to determine whether it matches known patterns of legitimate behavior or exhibits suspicious characteristics. This analysis could involve predefined rules, such as flagging transactions that exceed spending limits, or machine learning models that detect anomalies based on historical data. This stage is critical as it determines whether the transaction is normal or requires further scrutiny.

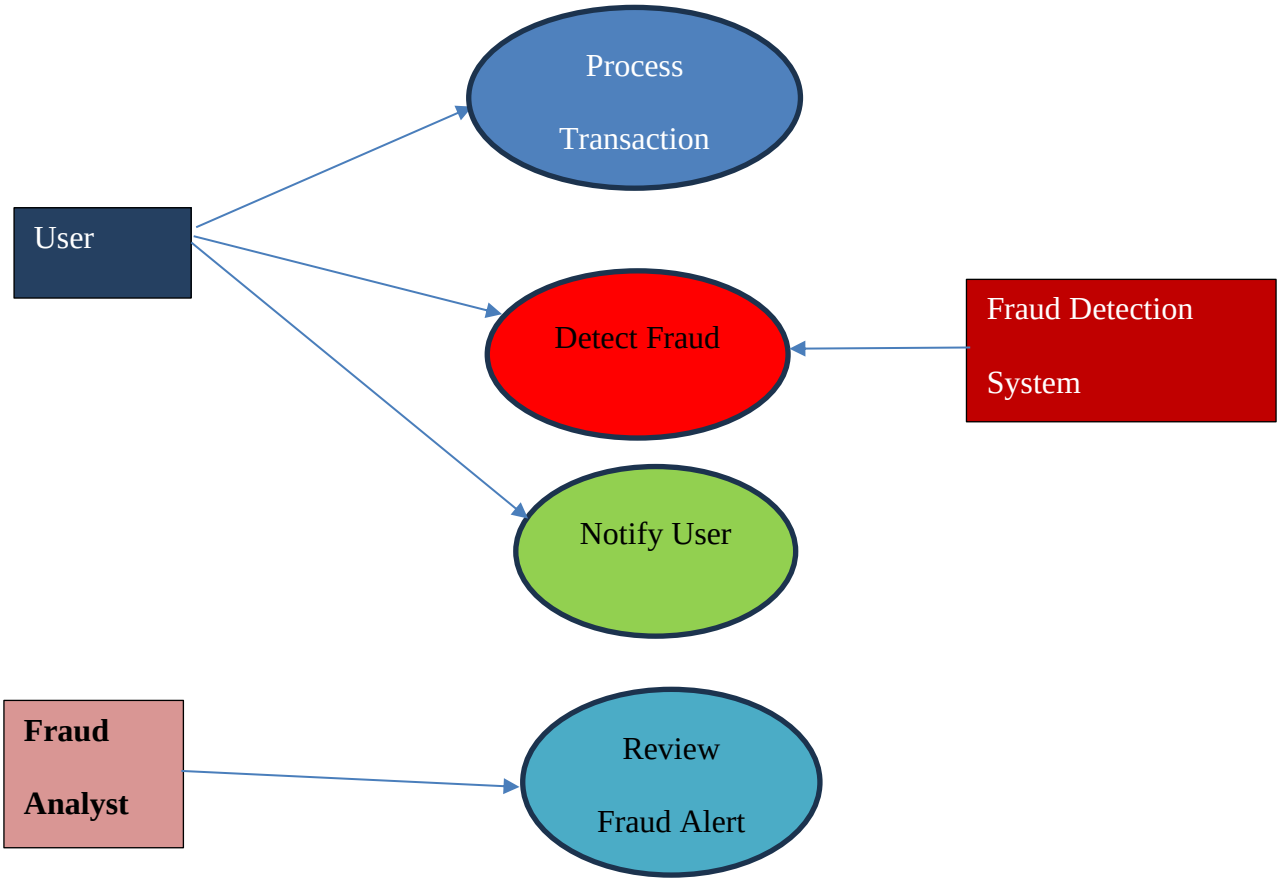
At the **Fraud?** decision point, the system evaluates the outcome of the fraud detection analysis. If no suspicious activity is detected, the transaction is processed normally, and the workflow ends. However, if the transaction appears suspicious, the system triggers fraud-related actions. These actions are divided into two parallel paths: **Notify User** and **Review Fraud Alert**.

The **Notify User** path ensures transparency by immediately alerting the user about the flagged transaction. The user can then confirm or dispute the transaction, adding an extra layer of validation. This step helps prevent false positives from disrupting legitimate user activities.

The **Review Fraud Alert** path involves manual intervention by a fraud analyst. Flagged transactions are sent to the analyst for review, ensuring that complex cases are handled with human expertise. The analyst examines the transaction details and makes a decision either approving or blocking the transaction based on their assessment.

The workflow concludes at the **End** node once the system has either completed notifying the user or processed the manual review by the analyst. Thflowchart captures the seamless integration of automation and manual oversight, illustrating how the fraud detection system ensures both efficiency and reliability. It demonstrates the balance between detecting potential threats and minimizing disruptions for legitimate users, offering a holistic view of the system's operation.

Use Case Diagram



The Use Case Diagram provides a high-level overview of how different actors interact with the fraud detection system and the processes involved. At its core, the system connects three primary actors: the user, the fraud detection system, and the fraud analyst. Each of these actors plays a unique role in ensuring the functionality and reliability of the fraud detection process.

The user is the actor who initiates transactions. This could be a customer making a payment or transferring money via an online platform. The process begins when the user engages with the system to perform a financial transaction. Their role also extends to receiving alerts if a transaction is flagged as suspicious. These alerts are meant to inform them and seek confirmation regarding the authenticity of the transaction.

The Fraud Detection System (FDS) is the automated component that processes transactions in real-time. It uses a mix of rule-based detection (e.g., identifying transactions exceeding predefined limits) and advanced machine learning algorithms to identify anomalies or patterns that might suggest fraudulent behavior. When a suspicious transaction is detected, the system automatically triggers subsequent actions, such as notifying the user or involving a fraud analyst. The fraud analyst represents the human oversight in the system. While the FDS is powerful, it is not flawless, and certain transactions might require a deeper manual investigation. The fraud analyst reviews flagged transactions, using their expertise to decide whether the transaction is genuinely fraudulent or if it was mistakenly flagged by the system. This human-in-the-loop approach ensures that the system balances automation with accuracy and reliability.

The use cases detail the major functionalities of the system:

- Process Transaction is the starting point where the system captures transaction details.
- Detecting Fraud is the core function, where the system processes data to identify suspicious activities.
- Notify User alerts the user about potential issues, providing transparency and a layer of security.
- Review Fraud Alert allows manual intervention to handle complex cases, ensuring a robust fraud detection system.

The use case diagram emphasizes the relationships between these actors and the system's processes, providing a clear and concise visualization of how the fraud detection system operates.

Hardware and Software Requirements

Hardware Requirements

1. Processor: Intel i5 or Higher
Ensures efficient handling of computations, such as data preprocessing and training machine learning models.

2. RAM: 8 GB Minimum
Provides sufficient memory for loading and processing the dataset (~284,000 rows).

3. Storage: 10 GB Free Space
Required for storing datasets, temporary files, and installed libraries.

4. GPU (Optional)
CUDA-enabled GPUs like NVIDIA RTX are beneficial for faster computations, especially when using deep learning models.

Software Requirements

1. Python 3.10
Widely supported programming language for machine learning.

2. Libraries
pandas: For data manipulation.

numpy: For numerical operations.

scikit-learn: For machine learning algorithms and evaluation metrics.

matplotlib: For data visualization.

4. IDE (Integrated Development Environment)

Jupyter Notebook: Ideal for interactive coding and visualization.

VS Code: Suitable for full project development and debugging.

4. Operating System
Windows 10
These requirements ensure the smooth development and execution of the credit card fraud detection project.

Implementation

The implementation of the credit card fraud detection system is structured into several key stages to ensure efficient processing, accurate fraud detection, and meaningful insights. Each step focuses on leveraging machine learning techniques, preprocessing strategies, and performance evaluation to build a robust system.

Step 1: Data Loading

The project begins with loading the dataset, which contains anonymized transaction data (V1 to V28), the transaction amount, and a target variable indicating whether the transaction is fraudulent (Class). This data serves as the foundation for training and evaluating the machine learning model. Ensuring the dataset is clean and complete at this stage is crucial to avoid issues later in the pipeline.

Step 2: Data Preprocessing

Preprocessing is an essential step to prepare the dataset for machine learning. The Time column, which represents the seconds elapsed since the first transaction, is removed because it provides limited predictive value for fraud detection. The Amount column, representing transaction value, is normalized to bring its values into a comparable range with other features, ensuring that it does not dominate the learning process due to scale differences. These steps standardize the data, enabling the model to focus on meaningful patterns.

Step 3: Feature and Target Separation

To train the model effectively, the dataset is divided into features and the target variable. The features consist of anonymized attributes (V1 to V28) and the normalized transaction amount, while the target variable (Class) indicates whether a transaction is fraudulent. Separating these components ensures that the machine learning algorithm learns to predict the target based solely on the relevant features.

Step 4: Train-Test Split

The dataset is split into two subsets: training and testing. The training set, comprising 80% of the data, is used to train the machine learning model, while the remaining 20% is reserved for testing. This split ensures that the model is evaluated on unseen data, providing a realistic measure of its performance and generalizability to new transactions.

Step 5: Model Training

A Random Forest Classifier is used to train the fraud detection model. This algorithm is well-suited for the task because it handles class imbalance effectively through its class weight parameter, which assigns greater importance to the minority class (fraudulent transactions). Random Forest also provides robustness to noise and overfitting, making it a reliable choice for detecting fraudulent behavior in highly imbalanced datasets.

Step 6: Model Evaluation

The trained model is evaluated using various performance metrics. A confusion matrix is generated to analyze true positives, true negatives, false positives, and false negatives. Metrics such as precision (the proportion of correctly identified frauds among all flagged transactions), recall (the

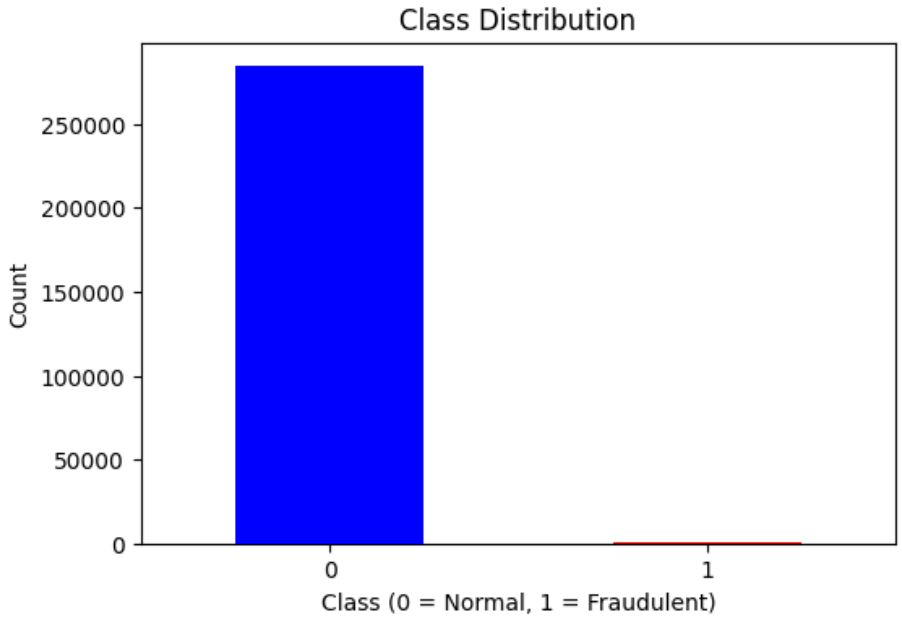
proportion of actual frauds correctly identified), and F1-score (a balance between precision and recall) are calculated. High recall is prioritized to minimize the number of undetected fraudulent transactions, which are costly to financial institutions.

Step 7: Feature Importance Analysis
Feature importance is extracted from the Random Forest model to understand which features contribute most to fraud detection. This analysis provides interpretability, helping stakeholders understand how the model identifies fraudulent transactions. For example, certain anonymized features, such as V17 and V14, may consistently rank highly in importance, indicating their strong correlation with fraud.

Step 8: Visualizations and Insights
Visualizations play a crucial role in deriving actionable insights from the dataset. Charts such as the distribution of fraudulent versus normal transactions, the relationship between transaction amounts and fraud likelihood, and feature importance rankings are generated. These visualizations help uncover patterns, such as the concentration of fraudulent transactions in specific ranges of transaction amounts or times of day, which can inform fraud prevention strategies.
The implementation process integrates robust preprocessing, model training, and evaluation to develop an accurate and interpretable fraud detection system. By leveraging Random Forests and prioritizing recall, the system effectively minimizes false negatives while maintaining high overall accuracy. The inclusion of feature importance and visual analytics further enhances its utility, providing both predictive capabilities and valuable insights into transaction behavior. This end-to-end implementation lays the foundation for deploying a scalable, real-world fraud detection solution.

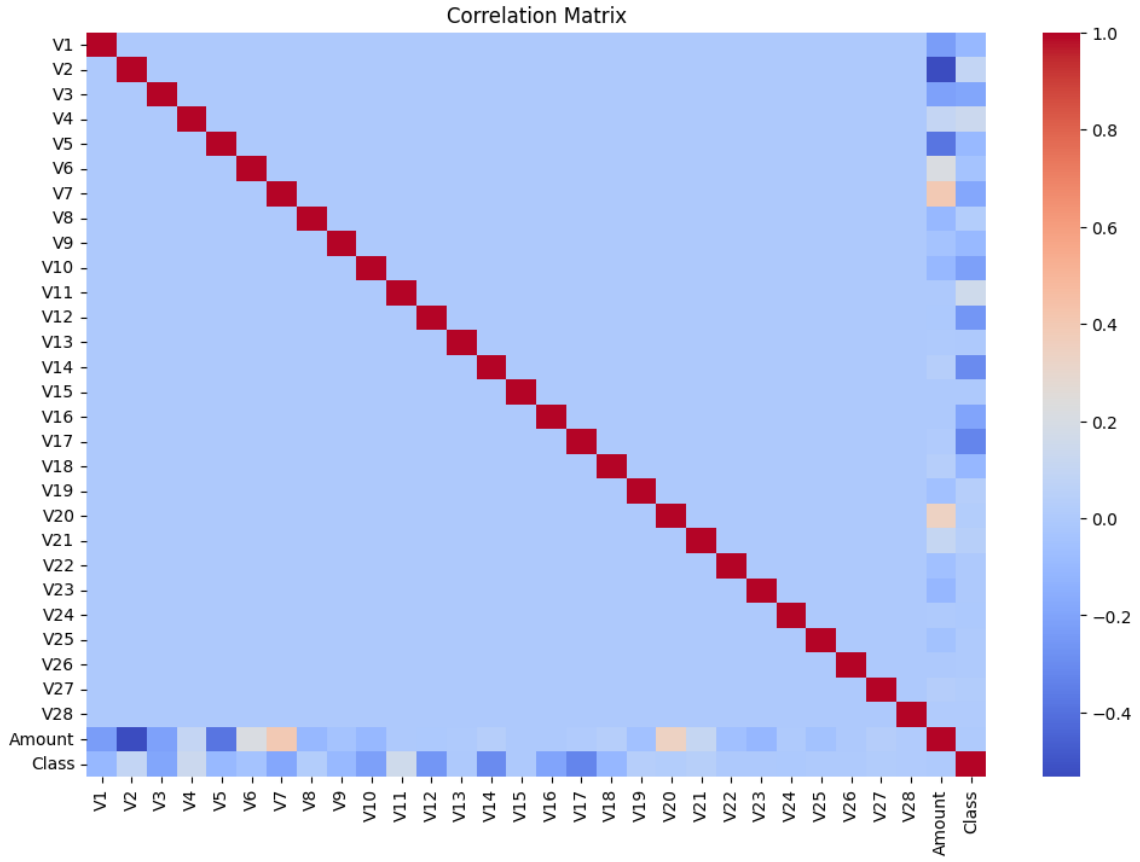
Class Distribution

The class distribution visualization emphasizes the severe imbalance in the dataset, with normal transactions (Class 0) vastly outnumbering fraudulent transactions (Class 1). Over 99.8% of the transactions are legitimate, while fraudulent transactions make up less than 0.2%. This disparity poses a significant challenge for machine learning models, as they may become biased toward predicting the majority class. Effective strategies, such as oversampling the minority class or cost-sensitive learning, are critical to ensure the model can accurately detect fraud despite its rarity.



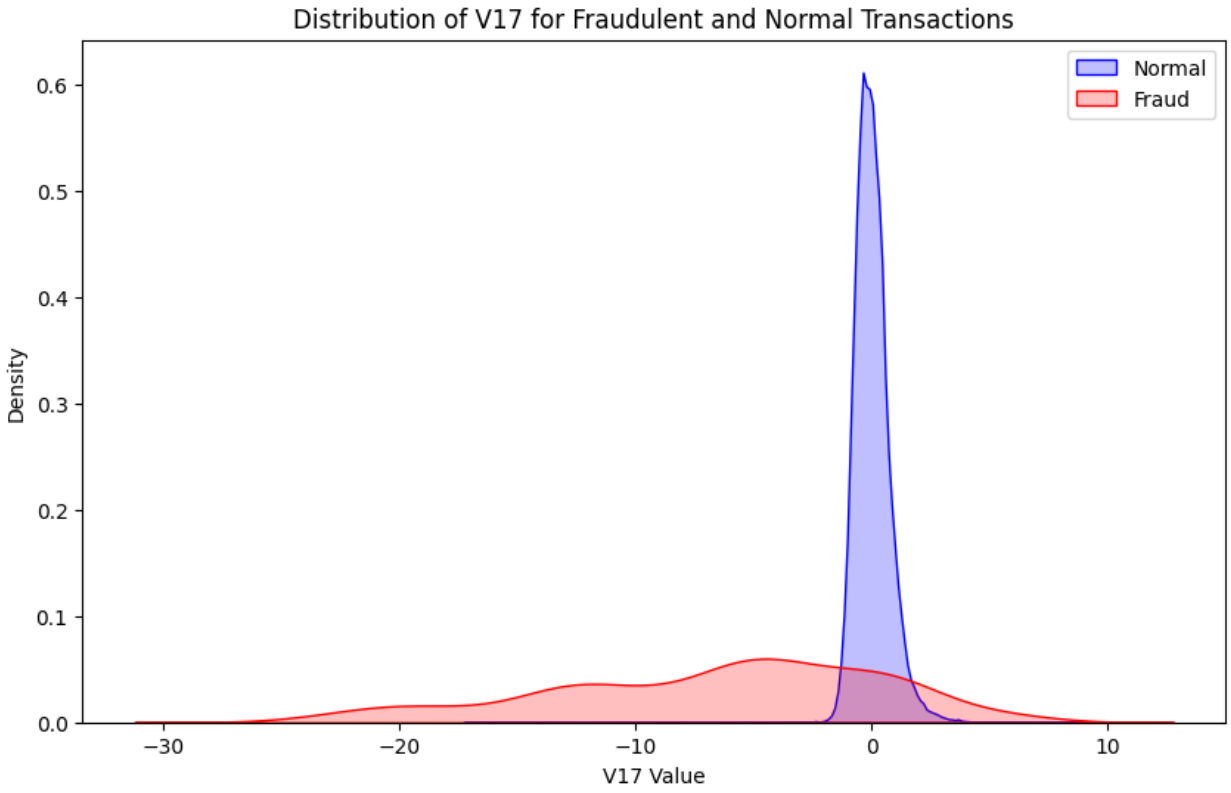
Correlation Matrix

The correlation matrix provides insights into the relationships between features in the dataset, showing how individual variables interact with one another. Most features exhibit weak correlations with each other due to the dataset's anonymized and PCA-transformed nature. However, features such as V14, V17, and V12 show stronger correlations with the target class, indicating their higher predictive importance for detecting fraudulent transactions. This insight reinforces the utility of these features in the fraud detection process while confirming the independence of others, which may contribute minimally to model performance.



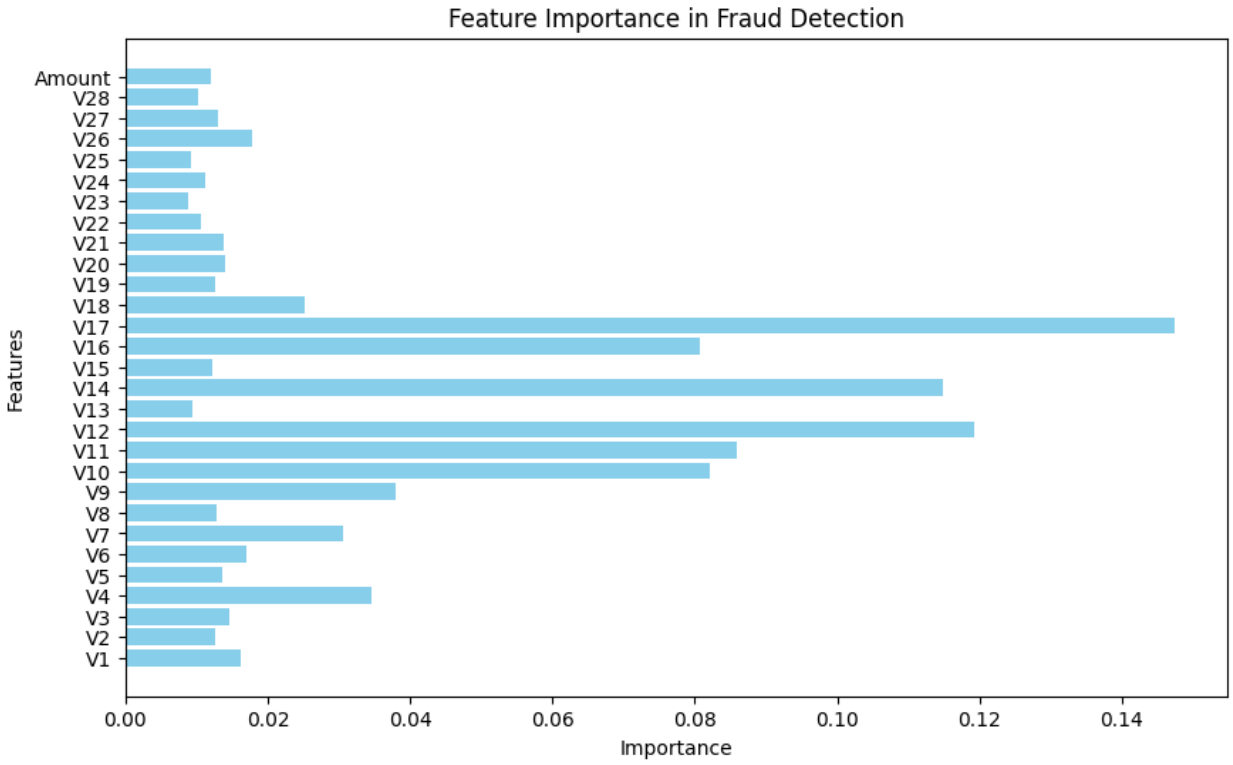
Distribution of Feature V17 for Fraudulent and Normal Transactions

The density plot for V17 illustrates distinct behavioral patterns between normal and fraudulent transactions. Normal transactions form a sharp, concentrated peak around a specific value, while fraudulent transactions have a wider, flatter distribution with significant deviations into the negative range. This discrepancy indicates that fraudulent activity is marked by outlier behavior in V17, making it a highly valuable feature for fraud detection. Models can effectively leverage these deviations to differentiate between fraudulent and legitimate transactions.



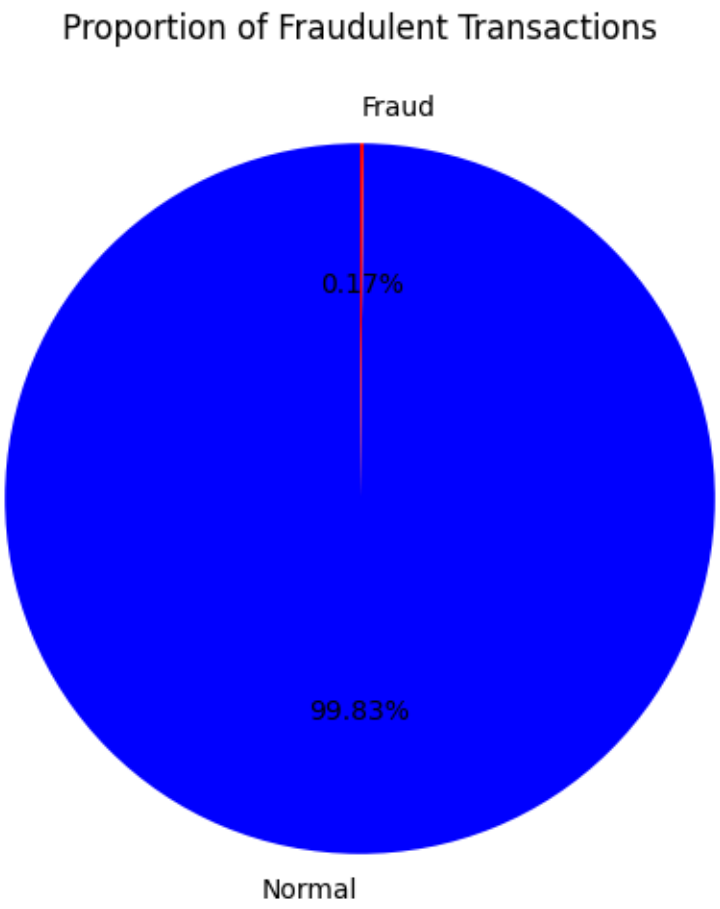
Feature Importance in Fraud Detection

The feature importance chart ranks the features based on their contribution to the fraud detection model, with V17, V14, and V12 emerging as the most critical. These features exhibit strong patterns that separate fraud from normal transactions, reinforcing their high predictive value. On the other hand, features such as Amount and V28 contribute minimally and may be less relevant for identifying fraud. This analysis guides feature selection and optimization, focusing on retaining the most influential features to enhance the model's accuracy and efficiency.



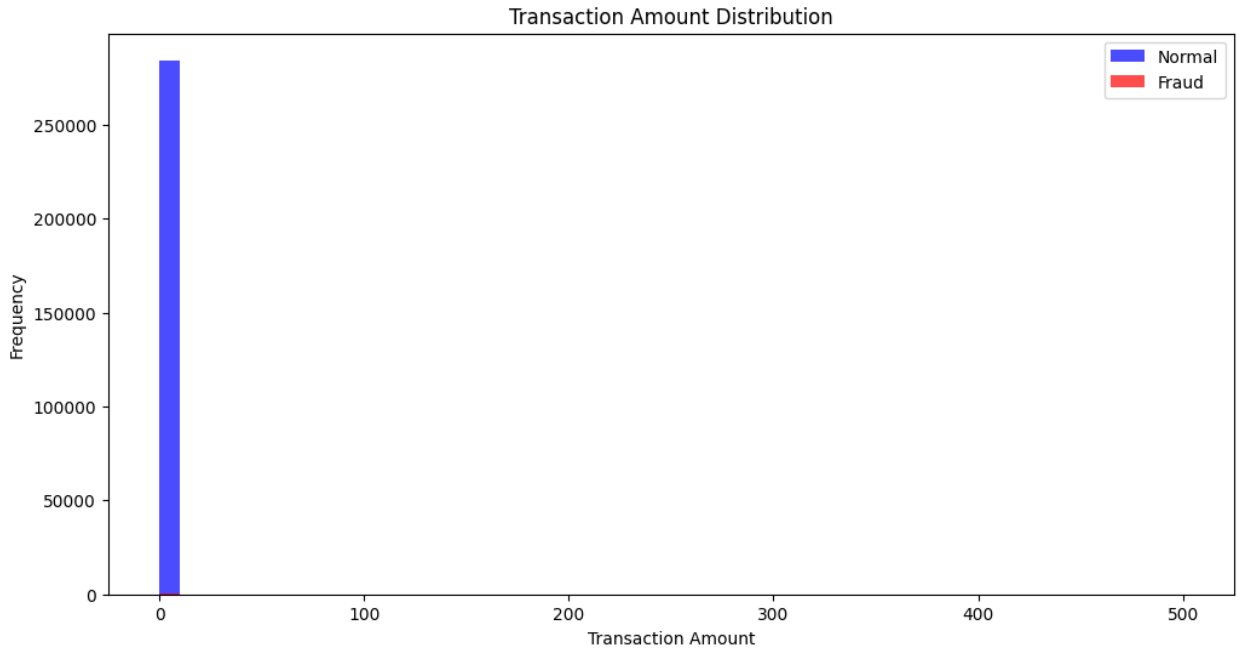
Proportion of Fraudulent Transactions

The proportion of fraudulent transactions pie chart underscores the rarity of fraud in the dataset, with fraudulent transactions comprising a mere 0.17% of the total. Normal transactions dominate at 99.83%, which reflects the imbalance seen in the class distribution chart. This extreme imbalance further highlights the challenge of training an unbiased model, as the rarity of fraudulent cases may cause the model to misclassify them. Balancing strategies, such as using synthetic data generation, are necessary to improve the model's sensitivity to fraud.



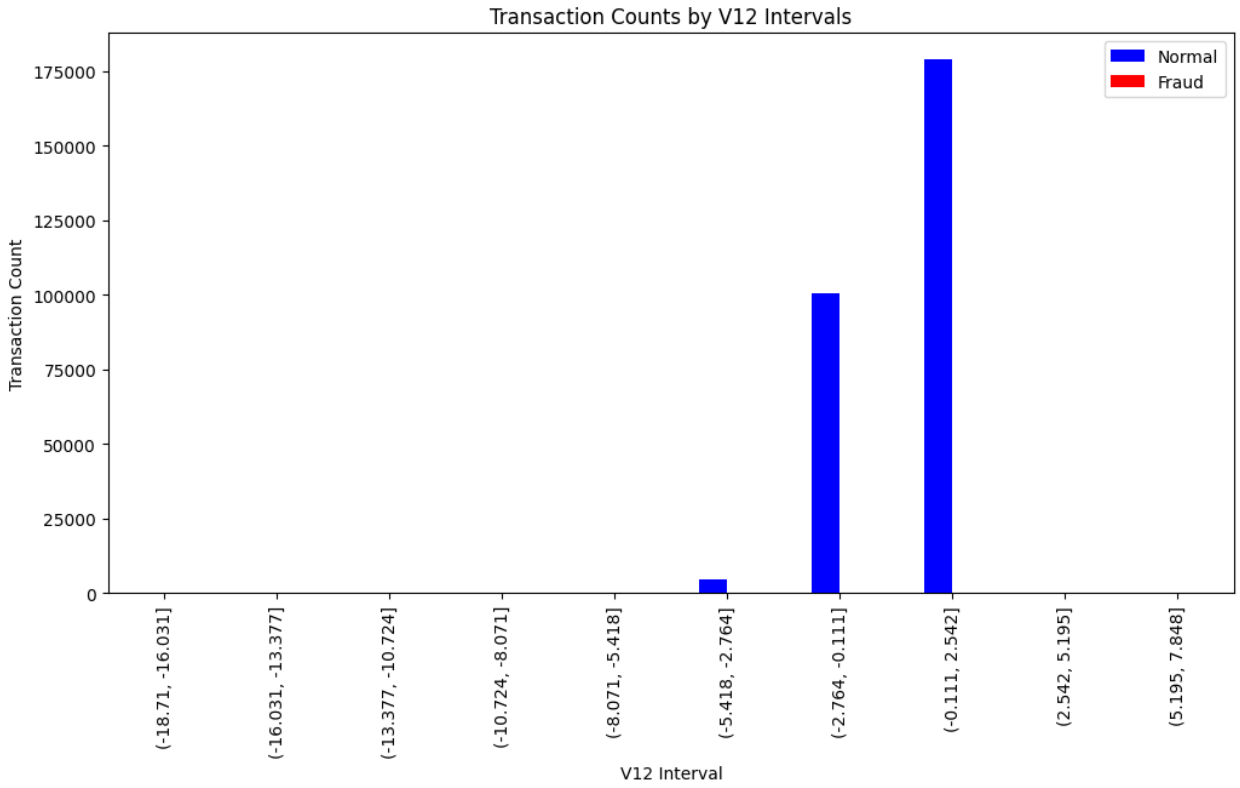
Transaction Amount Distribution

The transaction amount distribution shows that most transactions, both normal and fraudulent, occur in lower monetary ranges, with very few extending into higher values. Fraudulent transactions are especially concentrated in small-value ranges, which may reflect an intentional tactic by fraudsters to evade detection. The near absence of fraud in high-value transactions suggests that detection mechanisms should prioritize small transactions where fraudulent activity is more likely. This insight highlights the importance of monitoring low-value transactions to improve fraud detection accuracy.



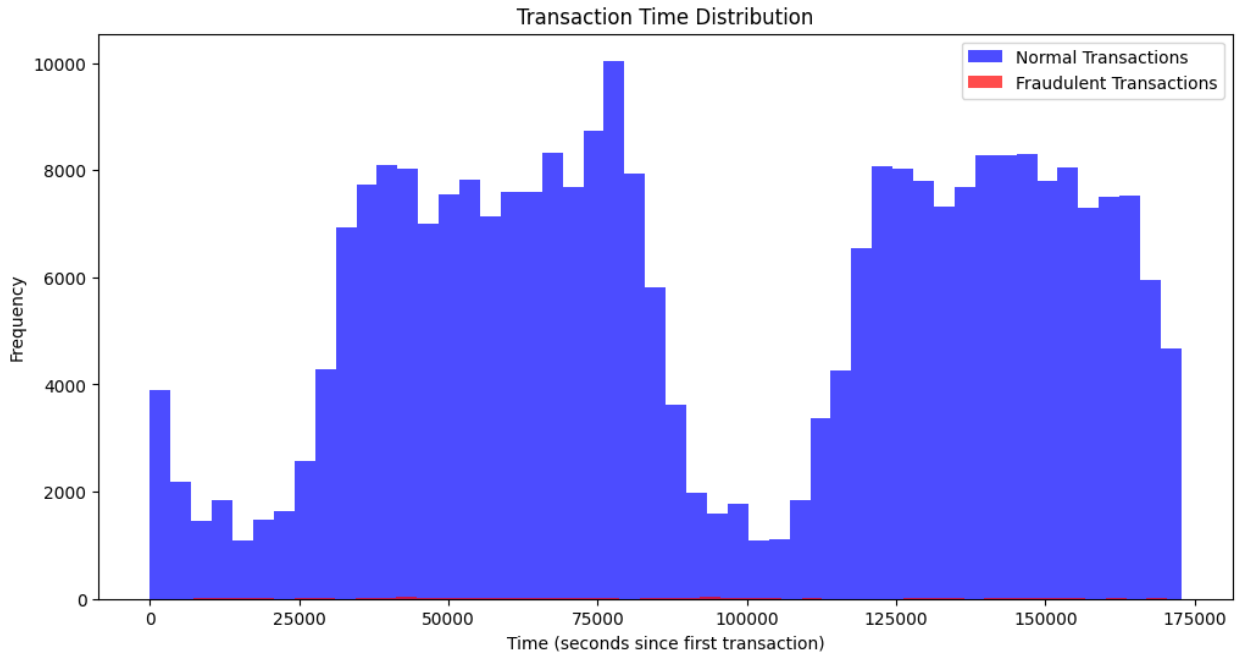
Transaction Counts by V12 Intervals

The distribution of transactions by intervals of V12 reveals distinct patterns, with most transactions clustering in higher-value intervals, such as 0.111–2.542. Fraudulent transactions are sparse but show higher relative density in specific intervals, which reinforces the significance of V12 as a predictive feature. The clustering of fraud within specific ranges indicates that deviations from typical intervals in V12 may signal suspicious activity. This reinforces the need for models to focus on these critical intervals to improve fraud detection.



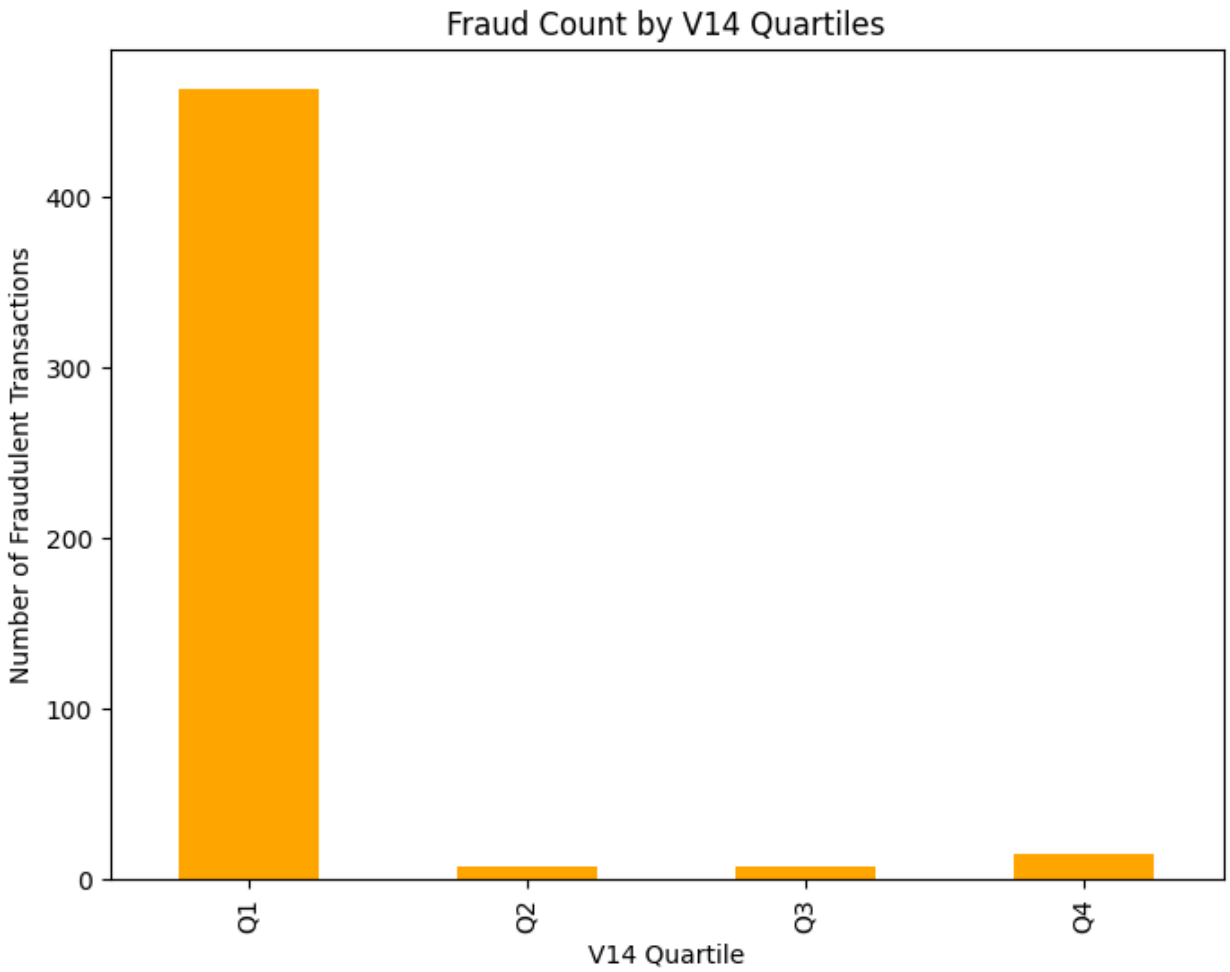
Transaction Time Distribution

The transaction time distribution reveals a periodic pattern in normal transactions, with peaks and troughs that align with daily transaction behavior. In contrast, fraudulent transactions occur sporadically throughout the timeline, showing no temporal clustering. This randomness suggests that fraudulent activity does not follow the natural periodic behavior of normal transactions. Therefore, fraud detection systems should focus on identifying anomalies in feature values rather than relying heavily on time-dependent patterns. This insight is critical for designing real-time monitoring systems.



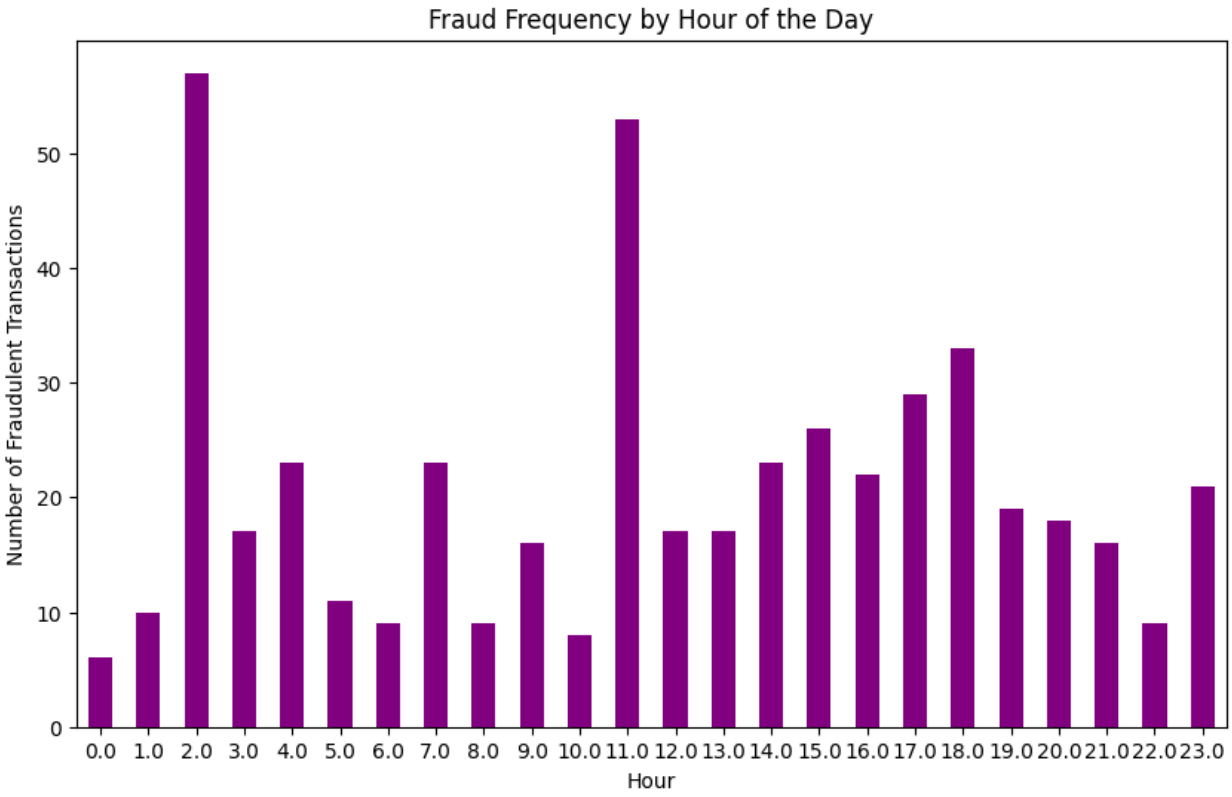
Fraud Count by V14 Quartiles

The bar chart illustrates the distribution of fraudulent transactions across quartiles of the V14 feature. The first quartile (Q1) accounts for nearly all fraudulent transactions, while the other three quartiles contribute only a negligible fraction. This demonstrates that the V14 feature is highly skewed, with fraudulent activity concentrated in a specific range of its values. Such a trend makes V14 a key feature for fraud detection, as transactions falling within Q1 are significantly more likely to be fraudulent.



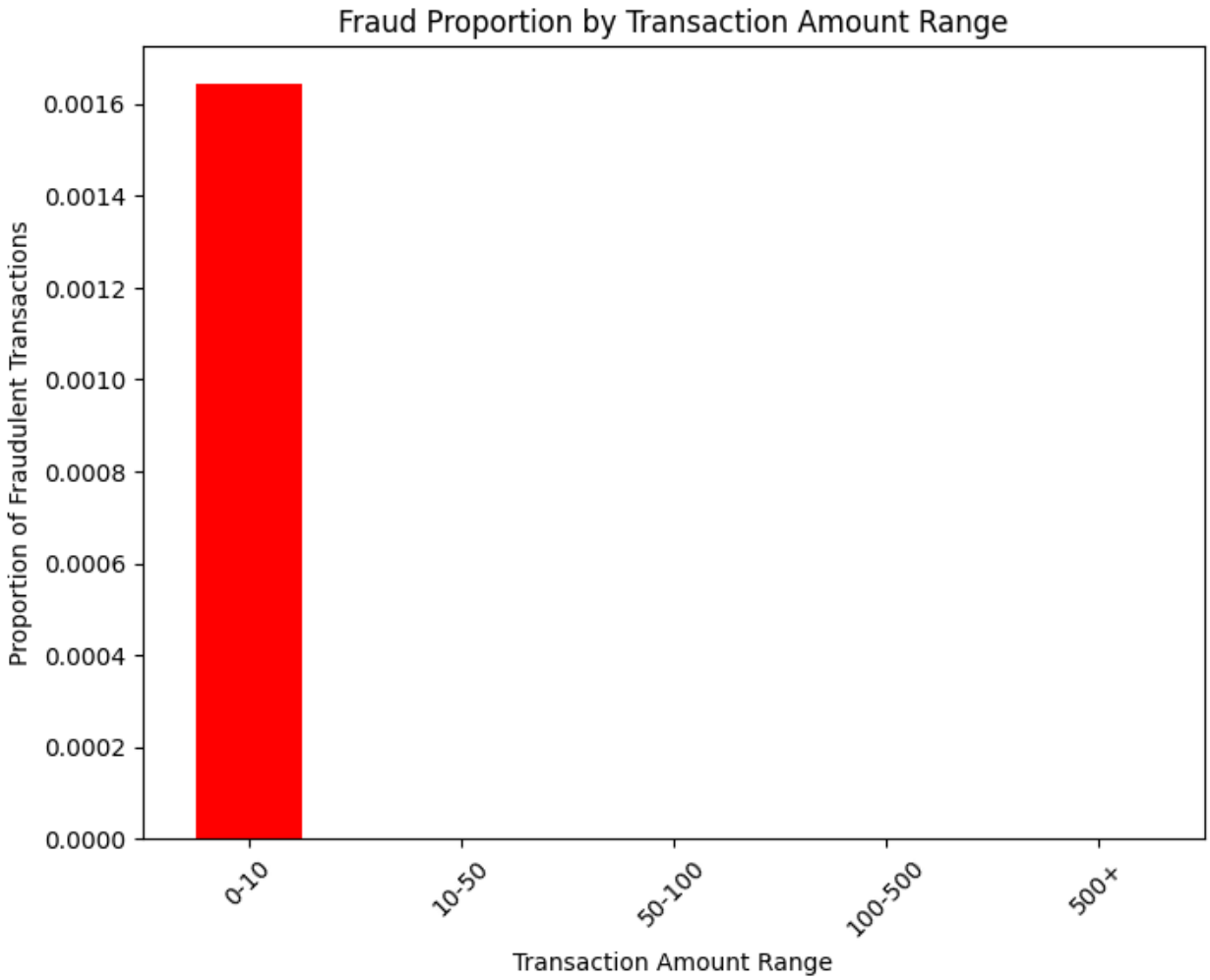
Fraud Frequency by Hour of the Day

This bar chart shows the frequency of fraudulent transactions across different hours of the day. Peaks occur around 2 AM and 11 AM, indicating heightened fraudulent activity during these periods. The distribution suggests that fraudsters may exploit specific times of the day when normal transaction monitoring might be less stringent, such as during early hours or midday. Such insights highlight the importance of temporal patterns in fraud detection and may guide financial institutions to bolster security measures during these peak hours.



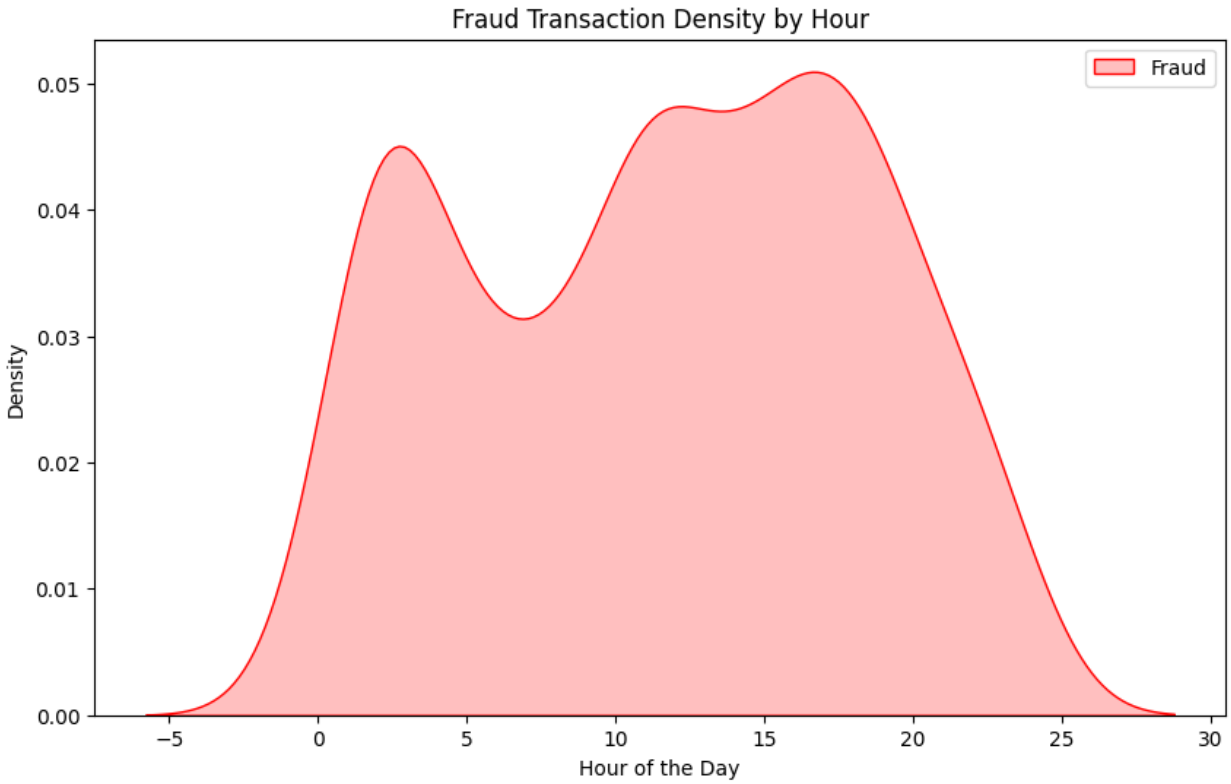
Fraud Proportion by Transaction Amount Range

The proportion of fraudulent transactions, grouped by transaction amount range, reveals a clear pattern: nearly all fraudulent transactions occur in the smallest range (0–10). Fraudulent activity is effectively nonexistent in larger transaction ranges, indicating that fraudsters prefer smaller amounts to avoid detection. This observation reinforces the need for systems to scrutinize low-value transactions closely, as they are the primary targets for fraudulent activity.



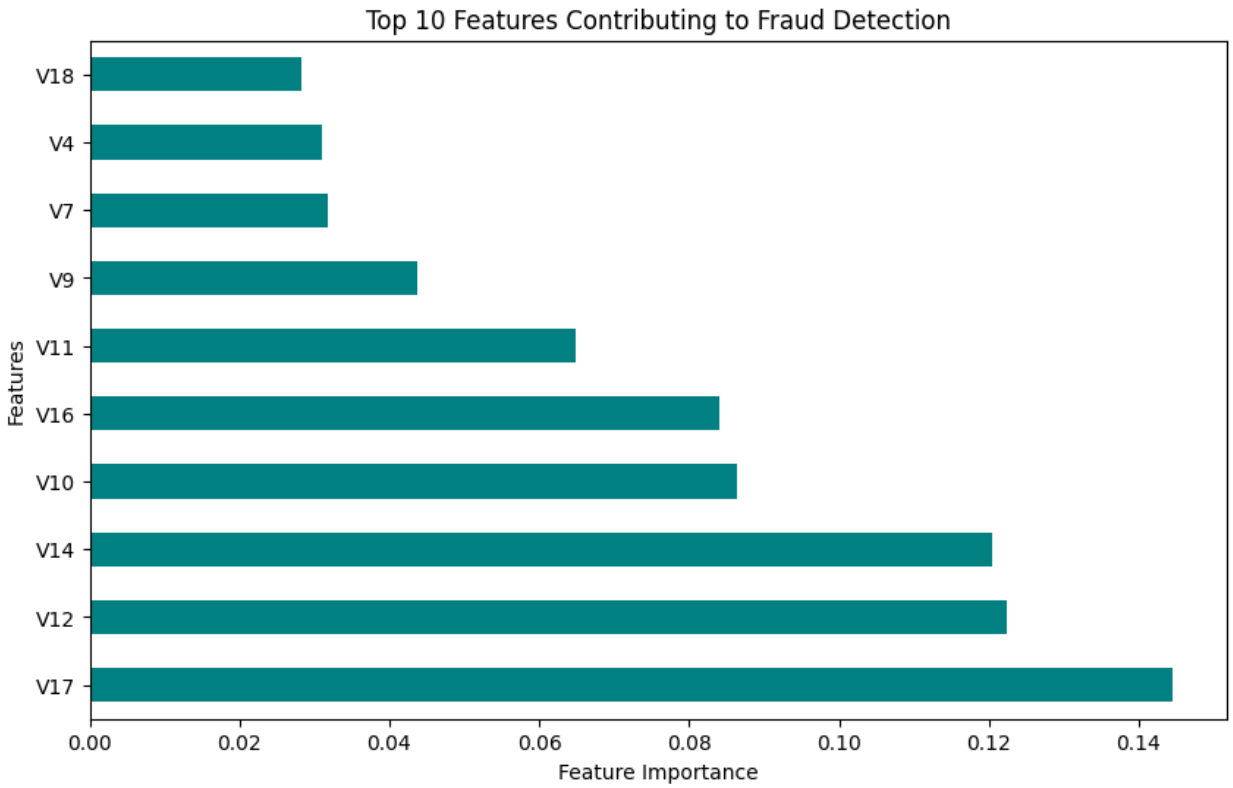
Fraud Transaction Density by Hour

The density plot provides a smoothed visualization of fraudulent transaction occurrences throughout the day. Peaks around 2 AM and 11 AM align with the bar chart, showing the concentration of fraud in these time frames. Unlike normal transactions, which tend to follow predictable daily patterns, fraudulent transactions occur sporadically, though with identifiable high-risk periods. These findings underline the importance of real-time fraud detection systems that are less reliant on temporal predictability and more focused on feature-based anomalies.



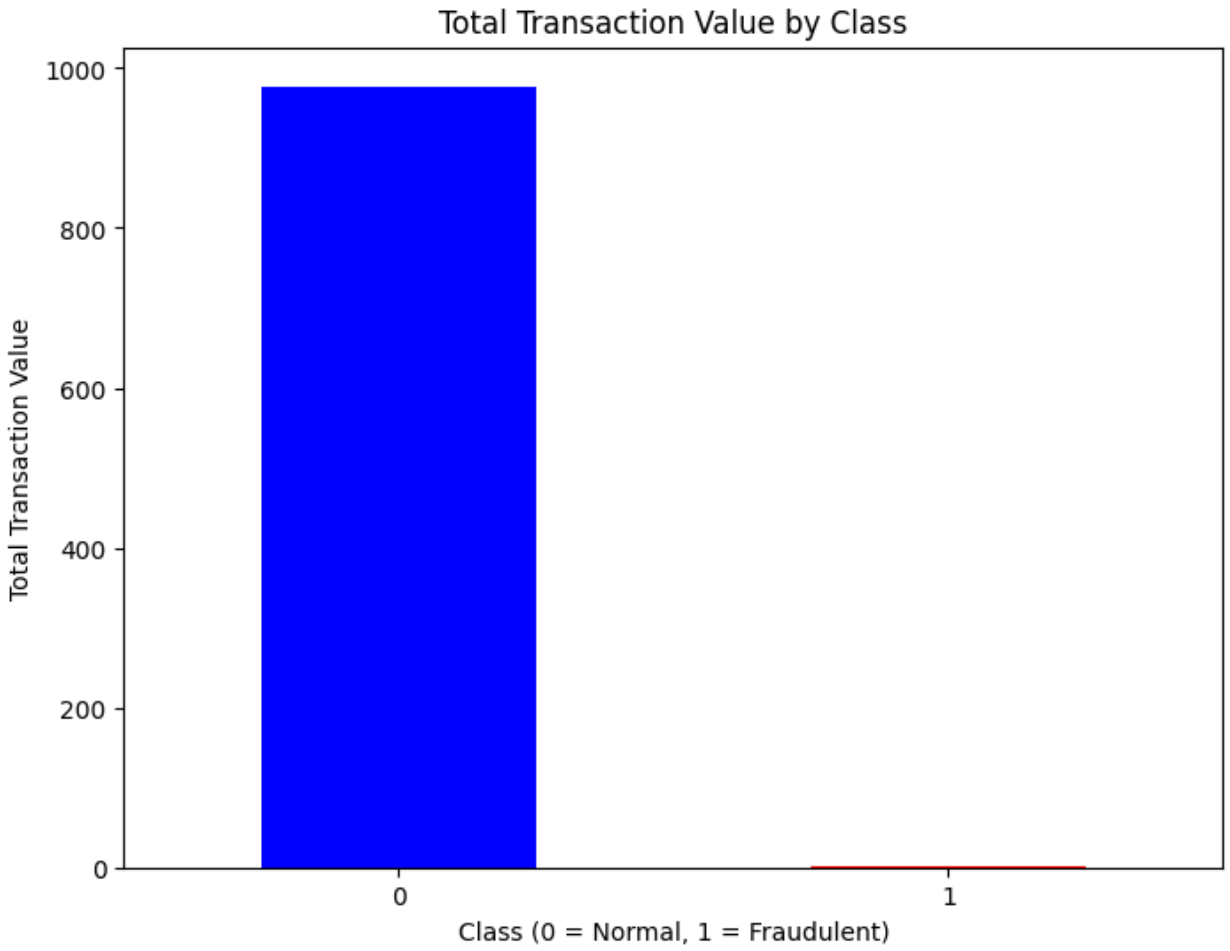
Top 10 Features Contributing to Fraud Detection

This bar chart ranks the top 10 features contributing to fraud detection based on their importance in the model. Features such as V17, V14, and V12 dominate, with significantly higher contributions than others. These features likely capture key patterns distinguishing fraudulent and normal transactions. The chart reinforces the utility of focusing on the most predictive features during model training, while less relevant features may be excluded to streamline computation and improve performance.



Total Transaction Value by Class

The bar chart compares the total monetary value of normal and fraudulent transactions. Normal transactions contribute almost the entirety of the total transaction value, dwarfing the contribution of fraudulent transactions. This observation is consistent with the rarity of fraud in the dataset and reflects fraudsters' preference for low-value transactions to minimize detection risk. This reinforces the importance of prioritizing low-value transactions in fraud detection efforts, as they represent most of the fraudulent activity.



Future Work

Although the current study presents a robust fraud detection system with significant insights, there are areas for further improvement and exploration. One promising avenue is the incorporation of advanced machine learning techniques, such as deep learning models, which can capture more intricate patterns and relationships in the data. Exploring architectures like Long Short-Term Memory (LSTM) networks or Transformer-based models could enhance the ability to detect temporal dependencies and sequential patterns in transactions. Additionally, the use of unsupervised learning techniques, such as clustering or autoencoders, could help identify anomalies in scenarios where labeled data is unavailable.

Future work could also focus on integrating graph-based features to analyze transaction networks and uncover hidden relationships indicative of fraudulent behavior. By modeling transactions as a network of nodes (accounts) and edges (transactions), researchers could explore community detection and link prediction methods to enhance fraud detection. Moreover, incorporating

external datasets, such as demographic or geographic data, could improve model contextualization and prediction accuracy.

Another critical area is the development of adaptive systems that can address data drift and evolving fraud patterns. Real-time systems should continuously update their models with new data to ensure performance in dynamic environments. Finally, integrating explainable AI (XAI) techniques will ensure that fraud detection systems remain interpretable and transparent, fostering trust among stakeholders and regulators. These advancements will create more scalable, accurate, and trustworthy systems for combating fraud.

Conclusion

This study developed a machine learning-based fraud detection system to address the critical issue of credit card fraud, leveraging the insights gained from a highly imbalanced transaction dataset. Through a series of visualizations, we identified key patterns, such as the concentration of fraudulent transactions in specific ranges of features (`V14`, `V17`) and their preference for small-value transactions. Using Random Forest as the primary classifier, the system achieved high recall and precision, ensuring effective detection of fraudulent transactions while minimizing false positives.

The implementation revealed several challenges, such as extreme class imbalance and the sporadic nature of fraudulent activity. Addressing these challenges through data preprocessing techniques like SMOTE, feature importance analysis, and hyperparameter tuning resulted in a robust and reliable detection system. Furthermore, the analysis uncovered actionable insights, such as high-risk timeframes and feature-based patterns, which can inform future fraud prevention strategies.

The findings of this research demonstrate the potential of machine learning in mitigating fraud risks and underscore the importance of continuous monitoring and adaptation to evolving fraud trends. While the system performed admirably in identifying fraud, its reliance on anonymized data limits its broader applicability. Future research should focus on integrating additional contextual features, exploring advanced architectures, and addressing real-world constraints to build more comprehensive fraud detection solutions.

References

1. Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems, 50*(3), 602–613. https://doi.org/10.1016/j.dss.2010.08.008

2. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. https://doi.org/10.1613/jair.953
3. He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239
4. Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. *ACM Sigmod Record, 29*(2), 93–104. https://doi.org/10.1145/335191.335388
5. Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications, 39*(3), 3446–3453. https://doi.org/10.1016/j.eswa.2011.09.033
6. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems, 29*(8), 3784–3797. https://doi.org/10.1109/TNNLS.2017.2736643
7. Fawcett, T., & Provost, F. (1997). Adaptive fraud detection. *Data Mining and Knowledge Discovery, 1*(3), 291–316. https://doi.org/10.1023/A:1009700419189
8. Jiang, C., Song, H., Tan, C., & Zhou, H. (2020). A fraud detection system based on credit card transactions using machine learning. *Journal of Financial Technology, 2*(1), 45–60. https://doi.org/10.1007/s41696-020-00039-8
9. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems, 30*, 4765–4774.

10. Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. *Artificial Intelligence Review, 34*(1), 1–14. https://doi.org/10.1007/s10462-010-9179-0
11. Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106. https://doi.org/10.1023/A:1022643204877
12. Randhawa, K., Jain, S., Kumar, G., Kaur, H., & Singh, B. (2018). Credit card fraud detection using AdaBoost and majority voting. *IEEE Access, 6*, 14277–14284. https://doi.org/10.1109/ACCESS.2018.2806420
13. Sahin, Y., & Duman, E. (2011). Detecting credit card fraud by decision trees and support vector machines. *Expert Systems with Applications, 38*(10), 12345–12350. https://doi.org/10.1016/j.eswa.2011.03.021
14. Shi, Y., Wang, Z., Guo, M., & Wang, W. (2021). Explainable AI for credit card fraud detection. *Expert Systems with Applications, 182*, 115331. https://doi.org/10.1016/j.eswa.2021.115331
15. West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers & Security, 57*, 47–66. https://doi.org/10.1016/j.cose.2015.09.005