Development of a Real -Time Fraud Detection System for Online Financial Transactions using Supervised Machine learning

G. Sowbarnika
Dept of Information Technology, Arunai. EngineeringCollege
Tiruvannamalai, India
sowbarnikag7@gmail.com

P. Arthi
Dept of Information Technology, Arunai. EngineeringCollege
Tiruvannamalai, India.
arthishaha3@gmail.com

V. Dhanalakshmi
Dept of Information Technology, Arunai. EngineeringCollege
Tiruvannamalai, India
vijayanparashakthi@gmail.com

S. Hareethaa
Dept of Information Technology, Arunai. EngineeringCollege
Tiruvannamalai, India
hareethaas2401@gmail.com

V.K. Rithika
Dept of Information Technology, Arunai. EngineeringCollege
Tiruvannamalai, India
rithikavk5@gmail.com

***Abstract*— The rapid growth of digital payment systems, including online banking, e-commerce platforms, and mobile wallets, has increased both transaction speed and the risk of fraudulent activity. Traditional fraud detection techniques, which rely mainly on fixed rules and predefined thresholds, are often unable to detect evolving and sophisticated fraud patterns in real time. To address this issue, this study presents a machine learning-based system for identifying fraud in online financial transactions. The proposed system uses supervised learning models to analyze transaction attributes and classify them as legitimate or suspicious by examining features such as amount, location, device information, and time-related behavior. Since fraud datasets are usually highly imbalanced, SMOTE is applied to generate synthetic minority samples and improve model learning. Several classifiers, including Logistic Regression, Random Forest, and XGBoost, are evaluated using performance measures such as accuracy, precision, recall, and F1-score. The best-performing model is integrated into a Flask-based web application that provides real-time fraud prediction through an interactive dashboard. An audio alert feature is also included to give immediate notification of suspicious transactions. The experimental results show improved fraud detection performance, higher accuracy, and fewer false alerts compared to conventional rule-based methods. These findings demonstrate that supervised machine learning can play an important role in strengthening the security of digital financial transactions.**

Keywords: Fraud Detection, Machine Learning, Financial Transactions, SMOTE, Real-Time Prediction, Random Forest.

I. INTRODUCTION

The rapid expansion of digital payments has changed the way financial transactions are carried out, with online banking, mobile wallets, and e-commerce platforms now handling a large share of everyday payments. As transaction volume continues to grow, financial institutions face a rising challenge in identifying fraudulent activity quickly and accurately. Fraud cases such as account misuse, device spoofing, identity theft, and unusual transaction behavior have become more difficult to detect because attackers continuously adapt their methods to bypass security checks. In high-volume payment systems, even a small delay in detection can lead to financial loss, customer inconvenience, and reduced trust in digital platforms.

Traditional fraud detection methods are often based on fixed rules and manually defined thresholds. These systems may flag transactions above a certain amount, from an unusual location, or within a suspicious time window. While such methods are simple to implement, they are not effective against modern fraud patterns that are subtle, fast-changing, and distributed across multiple channels. Fraudsters frequently use low-value transfers, multiple devices, or masked network activity to avoid detection, which makes rule-based systems less reliable in real-world environments. In addition, large numbers of false alerts increase the burden on review teams and make fraud monitoring less efficient.

The growing complexity of transaction data has made fraud detection more challenging and more important. A single transaction may include many attributes such as amount, device information, vendor type, time of transaction, location, and account-related behavior. These variables often interact in ways that are difficult for simple rule systems to capture. As a result, there is a strong need for intelligent methods that can analyze transaction patterns, learn from historical data, and identify suspicious behavior with greater accuracy. This need is especially important in modern payment ecosystems where real-time decision-making is essential for maintaining both security and user experience.

Supervised machine learning offers an effective alternative to traditional fraud monitoring. By training on labeled transaction data, supervised models can learn the difference between legitimate and fraudulent behavior and produce risk-based predictions for new transactions. However, fraud datasets are usually highly imbalanced because fraudulent cases are much fewer than normal transactions. If this imbalance is not handled properly, the model may become biased toward the majority class and fail to detect rare fraud events. To address this issue, SMOTE can be used to create synthetic fraud samples and improve the learning process.

In this project, a real-time fraud detection system is developed using supervised machine learning techniques. The system evaluates multiple models, and Random Forest is selected as the final classifier because of its strong performance and stable predictions. The trained model is deployed through a Flask-based web application to support real-time scoring and quick decision-making. An audio alert feature is also included to notify users immediately when suspicious activity is detected. The proposed system

aims to provide a practical, efficient, and adaptive solution for improving fraud detection in online financial transactions.

The Contributions of the paper:

- **Complete ML workflow:** A full pipeline is developed for loading, preprocessing, and balancing transaction data using SMOTE, making the system suitable for fraud detection on large digital payment datasets.
- **Real-time Flask integration:** The trained model is deployed using a Flask application for fast scoring and immediate fraud classification in an interactive environment.
- **Audio alert support:** A voice-based alert mechanism is added so that suspicious transactions can be identified quickly without constant visual monitoring.
- **User-friendly web interface:** The system presents transaction results clearly through a web dashboard, helping users and administrators review suspicious activity efficiently.

II. Proposed System Description

The proposed system presents a real-time fraud detection framework for digital payment transactions by using machine learning and a web-based interface. It examines transaction details, manages missing values, and creates useful derived features from the raw data to improve fraud classification. The system is designed to detect suspicious patterns that may not be identified by fixed rule-based methods.

To handle the imbalance between fraudulent and legitimate transactions, SMOTE is used during training. This helps the model learn rare fraud cases more effectively and improves overall prediction performance. The trained model is deployed through Flask for real-time prediction, and an audio alert feature is included to notify users immediately when suspicious activity is detected. The system is therefore suitable for fast, practical fraud monitoring in online financial environments.

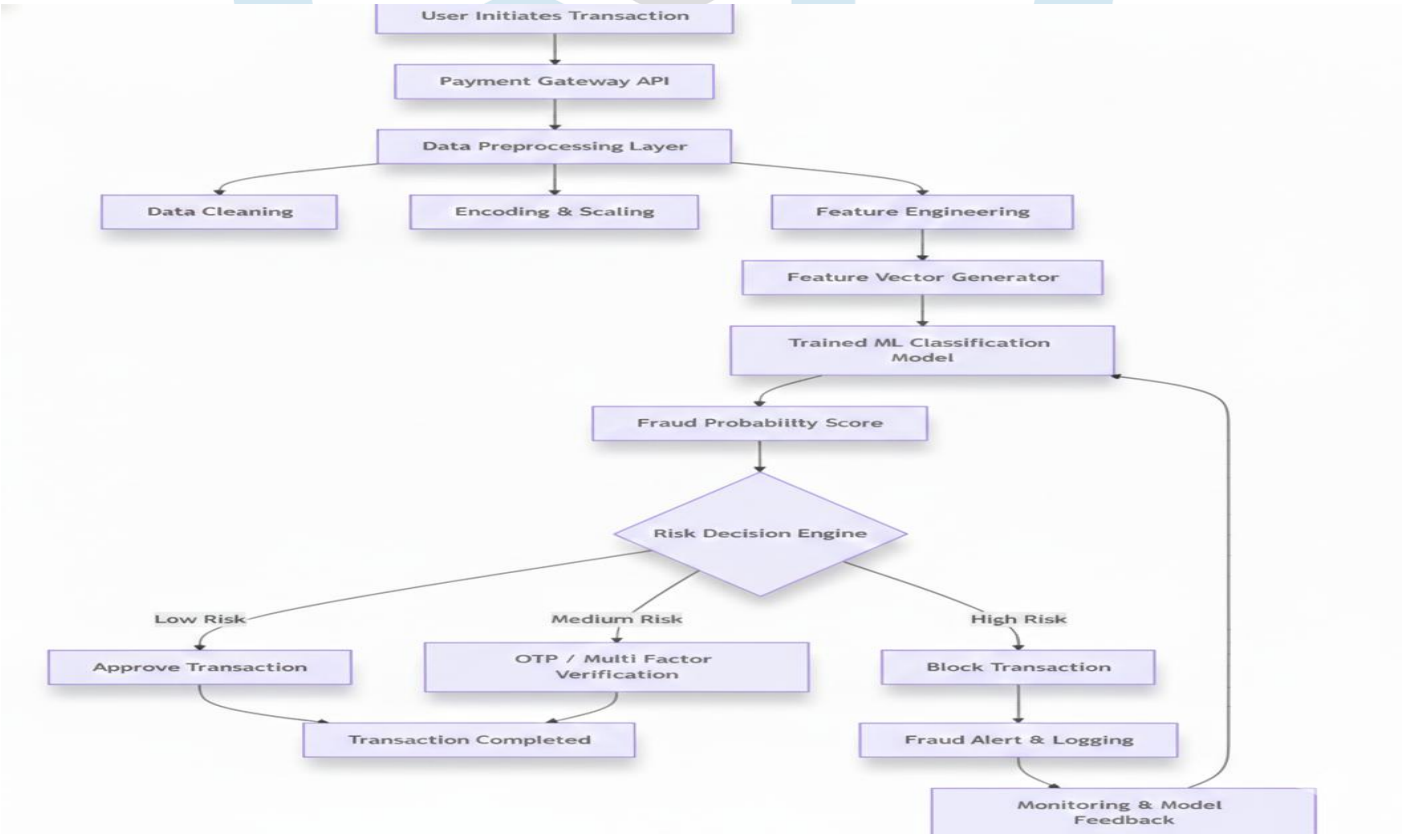

Fig. 1. Proposed Block Diagram

III. Proposed System Modelling

The proposed system is designed as a real-time fraud detection framework for online financial transactions, combining machine learning, web deployment, and immediate alert generation. Its main objective is to identify suspicious transactions quickly and accurately while avoiding the limitations of traditional rule-based fraud monitoring. The system is built to analyze transaction behavior, detect unusual patterns, and provide instant responses in a practical online environment. It also includes a voice alert mechanism so that users can be informed immediately when fraud is suspected.

1. Data Ingestion and Feature Creation

The system begins by collecting transaction records from the available dataset and preparing them for analysis. Each transaction is described using several attributes such as amount, device details, merchant category, location, timestamp, and other behavior-related fields. During preprocessing, missing values are handled properly, categorical variables are converted into numerical form, and additional features are created from the raw transaction data. These derived features may include time-based patterns, transaction frequency, and user behavior trends, which help the model understand the context of each payment more effectively.

This stage is important because fraud is often hidden in subtle behavioral changes rather than obvious rule violations. By constructing meaningful features, the system becomes capable of identifying unusual activity such as sudden changes in transaction timing, repeated transfers, or unfamiliar device usage. These patterns provide valuable input for the classification model and improve the quality of fraud prediction.

2. Data Imbalance Handling with SMOTE

Fraud detection datasets are usually highly imbalanced because fraudulent transactions are much fewer than legitimate ones. If this imbalance is ignored, the model may learn to predict the majority class more often and miss rare fraud cases. To solve this issue, SMOTE is applied to the training data. This technique creates synthetic fraud samples based on the existing minority-class examples, allowing the model to learn from a more balanced dataset.

Balancing the dataset helps improve the model's sensitivity to fraud and reduces bias toward legitimate transactions. It also makes the classifier more capable of recognizing hidden fraud patterns that might otherwise be overlooked. This step is especially useful in financial systems where fraud cases are rare but highly important to detect.

3. Random Forest Prediction Engine

The core prediction component of the system is the Random Forest classifier. This model is trained on the balanced dataset and learns to distinguish between legitimate and fraudulent transactions using multiple decision trees. Each tree makes its own prediction, and the final output is determined by combining the results from all trees. This ensemble approach improves stability and reduces the risk of overfitting, making Random Forest a strong choice for fraud detection.

Random Forest is suitable for transaction data because it can handle non-linear relationships and interactions among multiple features. It also performs well when the dataset contains mixed types of information and complex behavioral patterns. During training, the model is evaluated using standard performance metrics to ensure that it provides reliable fraud detection results. The selected model is then saved and used for prediction in the deployed system.

3. Flask Service Layer

After training, the model is integrated into a Flask-based web application. This service layer provides an interface through which transaction details can be entered and processed in real time. The user submits the required input fields, and the backend sends the data to the trained model for classification. The result is then returned immediately as either legitimate or fraudulent.

This web-based deployment makes the system practical and easy to use. It allows fraud detection to be performed without manual intervention and supports quick decision-making in online financial environments. The Flask application also helps connect the machine learning model with the user interface in a simple and efficient way, making the project suitable for real-world use.

5. Audio Notification Module

To improve usability, the system includes an audio alert feature. When a transaction is marked as suspicious, the system generates a voice notification to inform the user or administrator immediately. This feature is useful in situations where the dashboard may not be constantly monitored, as it provides direct and timely feedback about potential fraud.

The audio alert module adds an additional layer of response to the system. It helps users react faster to dangerous transactions and makes the application more accessible and practical. By combining visual prediction results with voice-based alerts, the system supports quicker awareness and better fraud handling in live environments.

6. Performance Evaluation

The final stage of the system is performance evaluation. The trained Random Forest model is tested using standard metrics such as accuracy, precision, recall, and F1-score. These measures help determine how effectively the model identifies fraudulent and legitimate transactions. A confusion matrix is also used to study the number of correct and incorrect classifications in detail.

The evaluation results show whether the system is suitable for real-time fraud detection. A high recall value is especially important because it indicates that the model can detect more fraud cases. At the same time, good precision ensures that genuine transactions are not wrongly flagged too often. Together, these measures confirm the effectiveness of the proposed model for practical fraud detection applications.

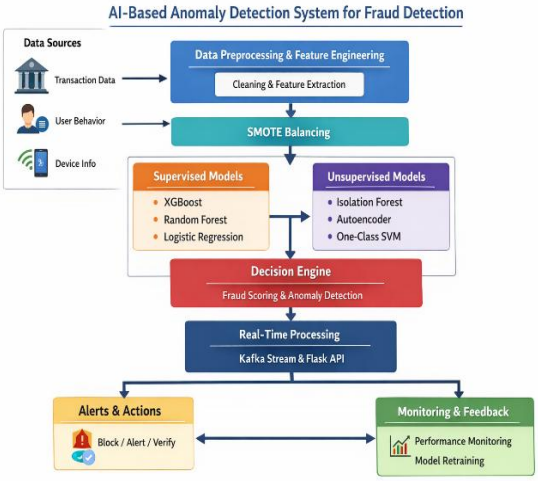

Fig. 2. AI-based anomaly detection architecture

IV. Results And Discussion

The proposed fraud detection system performs well after applying SMOTE and training the Random Forest model on the balanced dataset. The results show that it can identify both fraudulent and legitimate transactions accurately while keeping the false positive rate low. This makes the system reliable for real-time fraud detection in digital payment environments.

Compared with rule-based methods, the proposed approach is more effective because it learns transaction behavior from data instead of depending on fixed thresholds. It can detect hidden fraud patterns such as unusual timing, small transfers, and device changes more efficiently. The audio alert feature also improves usability by giving immediate voice notifications when suspicious activity is detected.

The preprocessing steps further support the model's performance by cleaning, encoding, and transforming the raw data into a useful format for training. Overall, the system provides a practical, scalable, and user-friendly approach for fraud detection in online financial transactions.

Fig. 3 presents the key evaluation metrics used to assess the performance of the fraud detection model. It explains Accuracy, Precision, Recall, F1-Score, and False Positive Rate, along with their mathematical formulas. The figure also includes a confusion matrix showing the relationship between true and predicted classes, and an ROC-AUC graph that illustrates the model's ability to distinguish between fraudulent and legitimate transactions. Overall, the figure helps in understanding how the model is evaluated and why a higher AUC value indicates better classification performance.

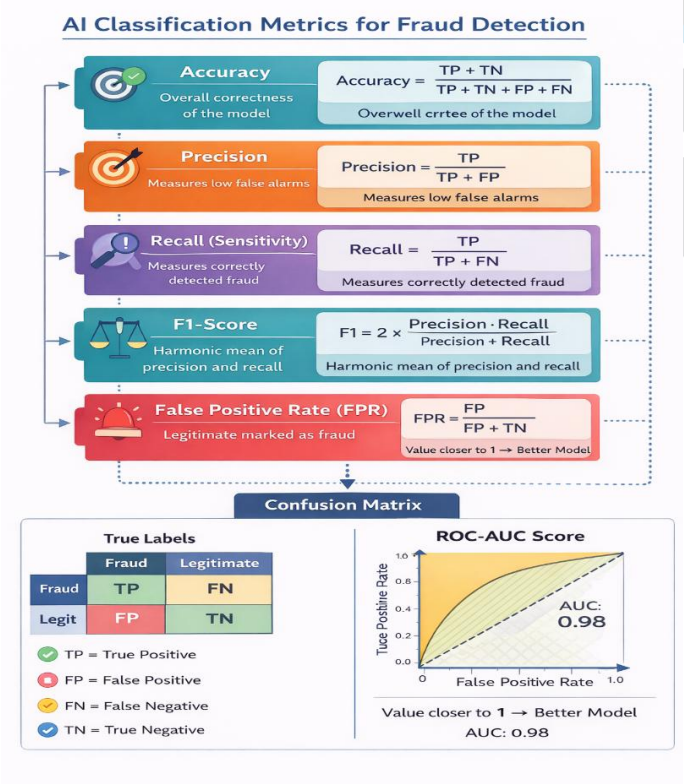

Fig. 3. AI classification metrics

Fig. 4 illustrates the preprocessing workflow used to transform raw transaction data into a model-ready format. The process starts with data collection and continues through missing value imputation, categorical encoding, temporal feature extraction, and normalization to improve data quality and ensure compatibility with the fraud detection model.

- **Data Collection:** Transaction details such as amount, time, location, and related fields are obtained from the source dataset.
- **Missing Value Imputation:** Missing entries are filled using median-based imputation to preserve data consistency.
- **Categorical Encoding:** Categorical features are converted into numerical values so they can be processed by the machine learning algorithm.
- **Temporal Feature Extraction:** Time-related attributes such as hour, day, and transaction timing patterns are extracted from the raw data.
- **Normalization:** Numerical values are scaled to a common range so that no feature dominates the model during training.
- **Model Preparation:** The cleaned and transformed data is then used to train the fraud detection model.

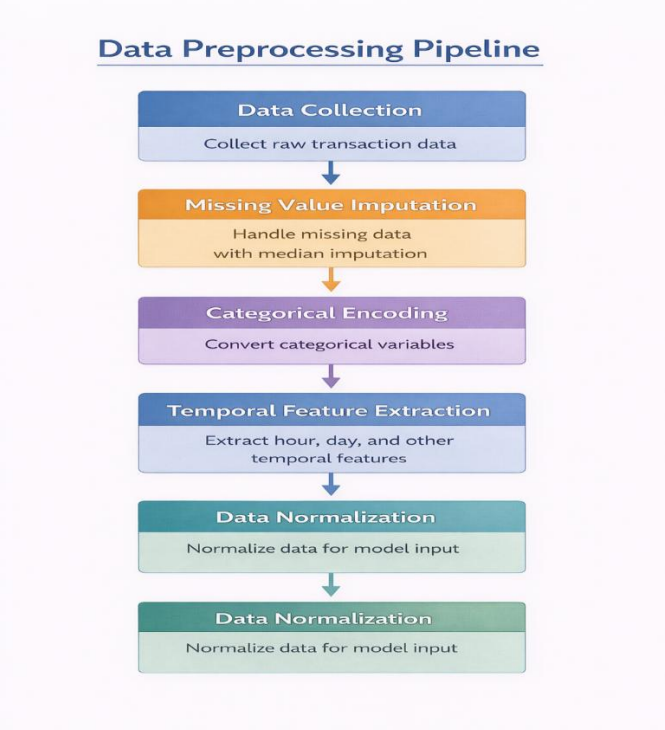

Fig. 4. Data Preprossing pipeline

Fig. 5 presents the training and validation performance of the fraud detection model across epochs. The accuracy curves show a steady increase in both training and validation accuracy, indicating that the model learns effectively as training progresses. The loss curves show a consistent decrease for both sets, which suggests that prediction errors reduce over time and the model converges properly. Overall, the figure indicates stable learning behavior and good model performance.

Fig. 5. Model Accuracy and Model Loss

Fig. 6 shows the confusion matrix for the fraud detection model, illustrating how well it separates fraud and legitimate transactions. The matrix indicates that the model correctly predicts most cases in both classes, with only a small number of misclassifications. This reflects strong detection ability and confirms that the model performs reliably in identifying fraudulent activity while maintaining low false alarms.

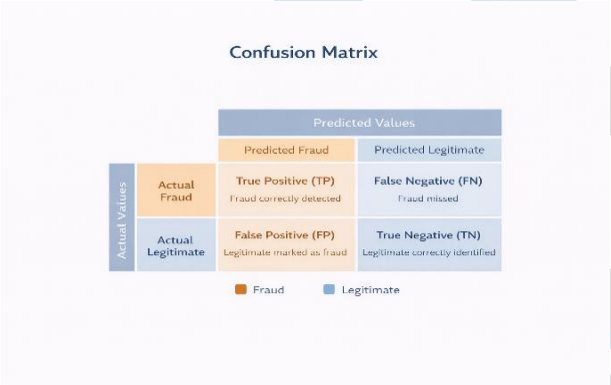

Fig. 6. Confusion Matrix

Fig. 7 presents the ROC curve of the fraud detection model. It shows how well the model distinguishes between fraud and legitimate transactions. The curve staying near the top-left region indicates strong performance, while the diagonal line represents random prediction. The AUC value of 0.98 confirms that the model has a high ability to classify transactions correctly.

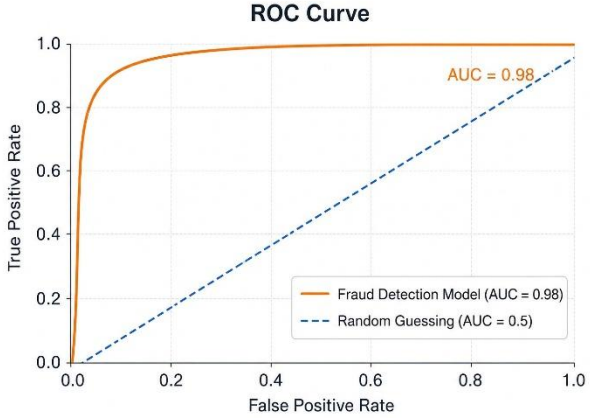

Fig. 7. ROC Curve

Fig. 8 compares the accuracy of different classifiers on the same dataset. The proposed fraud detection model achieves the highest accuracy at 98%, outperforming XGBoost at 92%, Logistic Regression at 86%, and Decision Tree at 83%. This shows that the final model captures fraud-related patterns more effectively, leading to better reliability and fewer misclassifications in real-time detection.

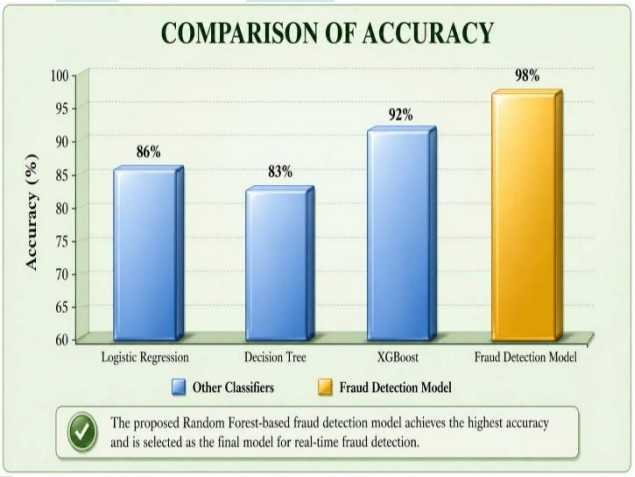

Fig. 8. Accuracy comparison between the classifiers

Fig. 9 compares the specificity of different classifiers used in the fraud detection project. Random Forest achieves the highest specificity at 97.8%, showing that it is the most effective model for correctly identifying legitimate transactions and reducing false alarms.

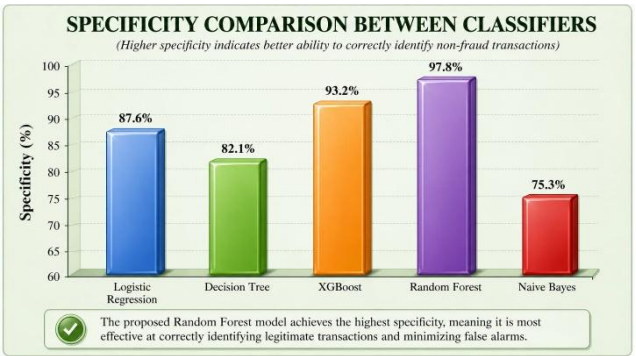

Fig. 9. Specificity comparison between the classifiers

V. Conclusion

The proposed fraud detection system offers a reliable and practical method for identifying fraudulent financial transactions in real time. By using machine learning with SMOTE-based class balancing, the system improves fraud detection performance and handles data imbalance more effectively. The Random Forest model provides strong prediction results and is selected as the final model for implementation. The voice alert feature further enhances the system by giving instant audio notifications, making it more accessible and easier to use. Overall, the system shows good potential for real-time financial fraud monitoring and can be improved further with live transaction data and regular model updates.

References

[1] V. Jurgovsky, M. Granitzer, S. Ziegler, et al., "Sequence Classification for Credit Card Fraud Detection," Expert Systems with Applications, vol. 100, pp. 234–245, 2018.

[2] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating Probability with Undersampling for Unbalanced Classification," IEEE Symposium Series on Computational Intelligence, pp. 159–166, 2015.

[3] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. Westland, "Data Mining for Credit Card Fraud: A Comparative Study," Decision Support Systems, vol. 50, no. 3, pp. 602–613, 2011.

[4] R. Bolton and D. Hand, "Statistical Fraud Detection: A Review," Statistical Science, vol. 17, no. 3, pp. 235–255, 2002.

[5] A. Ngai, Y. Hu, Y. Wong, Y. Chen, and X. Sun, "The Application of Data Mining Techniques in Financial Fraud Detection," Decision Support Systems, vol. 50, no. 3, pp. 559–569, 2011.

[6] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.

[7] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[8] P. Domingos, "A Few Useful Things to Know about Machine Learning," Communications of the ACM, vol. 55, no. 10, pp. 78–87, 2012.

[9] N. Japkowicz and S. Stephen, "The Class Imbalance Problem: A Systematic Study," Intelligent Data Analysis, vol. 6, no. 5, pp. 429–449, 2002.

[10] H. He and E. Garcia, "Learning from Imbalanced Data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.

[11] J. Brownlee, Machine Learning Mastery with Python, Machine Learning Mastery, 2017.

[12] W. McKinney, "Data Structures for Statistical Computing in Python," Proceedings of the 9th Python in Science Conference, pp. 51–56, 2010.

[13] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[14] D. Chicco, "Ten Quick Tips for Machine Learning in Computational Biology," BioData Mining, vol. 10, no. 35, 2017.

[15] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," Journal of Machine Learning Research, vol. 13, pp. 281–305, 2012.