IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15419

UPI Fraud Detection Using Hybrid Machine Learning Models with Explainable Risk Scoring and Real-Time Monitoring

Tarra Sekhar[1], G. Vijaya Kumar[2]

M. Tech Scholar, Computer Science and Engineering, Pragati Engineering College(A), Surampalem, India[1]

Assistant Professor, Computer Science and Engineering, Pragati Engineering College(A), Surampalem, India[2]

Abstract: Unified Payments Interface has become a major transaction channel in day-to-day digital payments, which makes transaction security an important technical concern. Fraud detection in such environments is difficult because suspicious transactions form only a very small portion of the total data, while legitimate user activity can vary widely in amount, timing, and transaction type. This work presents a hybrid machine learning framework for UPI fraud detection that combines a baseline Random Forest model with an improved XGBoost-based detection model, supported by imbalance handling, engineered transaction features, explainable prediction analysis, and a real-time monitoring dashboard. The system is designed as a complete end-to-end pipeline consisting of dataset preparation, preprocessing, feature transformation, model training, fraud scoring, API-ready prediction flow, and interactive visualization. In the implementation, transaction attributes are transformed into a compact feature set that includes temporal behaviour, transaction-category indicators, and fraud-rate information by transaction type. Synthetic Minority Over-sampling Technique is used to reduce the effect of class imbalance during training. The trained model produces a fraud probability score, and a decision threshold of 0.6 is used for final classification. To improve transparency, SHAP-based feature explanations are integrated so that important factors behind a prediction can be viewed instead of treating the model as a complete black box. A Streamlit dashboard was developed to support manual fraud checking, risk-gauge visualization, live transaction simulation, feature-importance display, and monitoring analytics. Experimental results show strong discrimination between legitimate and suspicious transactions. From the recorded confusion matrix, the system correctly identifies 314224 legitimate transactions and 168 fraudulent transactions, with zero false positives and 181 false negatives. These results indicate that the proposed framework is highly reliable for safe-transaction confirmation and practically useful for risk-aware UPI monitoring applications. The work demonstrates that combining ensemble learning, explainability, and dashboard-based monitoring can provide a more usable fraud detection system than rule-based screening alone.

Keywords: UPI fraud detection, XGBoost, Random Forest, Explainable AI, SHAP, SMOTE, Streamlit dashboard, financial transaction security

I. INTRODUCTION

Digital payment systems are now a regular part of financial activity, and in India the Unified Payments Interface has become one of the most widely used platforms for instant money transfer, merchant payment, and peer-to-peer transactions. The popularity of UPI comes from its speed, simplicity, and continuous availability. A user can initiate and complete a transaction within seconds using a mobile device and a UPI identifier. This convenience has improved digital accessibility for individual users, small businesses, students, and service providers.

At the same time, the growing use of UPI has widened the surface for fraudulent activity. Suspicious payment requests, unauthorized fund transfers, social-engineering attacks, phishing links, QR-code manipulation, and deceptive transaction patterns have become serious challenges in digital finance. Fraud detection in this domain is not a straightforward classification problem. A transaction may appear unusual and still be legitimate, while a fraudulent transaction may imitate the pattern of ordinary activity. This overlap makes simple threshold-based verification inadequate in many practical situations.

Traditional fraud monitoring systems are often based on manually designed rules. Such rules may block or flag a transaction when the amount is unusually high, the frequency is abnormal, or the transaction source differs from expected behaviour. Although these methods are simple to understand and easy to deploy, they are limited in adaptability. Fraud

DOI: 10.17148/IJARCCE.2026.15419

patterns do not remain fixed. Attackers continuously modify their strategies, and static rules often fail to capture such evolving behaviour. As transaction volume increases, manual review also becomes difficult to sustain.

Machine learning provides a stronger alternative because it can learn behaviour from historical transaction data rather than depend entirely on manually defined conditions. By examining multiple variables together, a model can identify patterns that are difficult to express using fixed rules. Ensemble methods are particularly suitable for structured financial data because they can handle feature interactions, categorical transformations, and non-linear decision boundaries with good predictive performance [1], [2].

The work presented in this paper develops a UPI fraud detection framework built around two ensemble models. Random Forest is used as the baseline model to establish initial predictive behaviour, while XGBoost is used as the improved model for stronger fraud discrimination [2], [7]. The system also incorporates class imbalance handling using SMOTE, explainable prediction support using SHAP, and a dashboard interface for human-readable fraud monitoring [8], [9]. In addition to model development, this work emphasizes usability, because a fraud detection model becomes more meaningful when its outputs can be interpreted and monitored through an interactive interface.

II. RELATED WORK

Financial fraud detection has been studied extensively in the broader context of banking, payment cards, e-commerce, and online transaction monitoring. Recent literature shows a gradual shift from rule-based screening to machine learning and hybrid intelligent methods. A systematic review by Ali et al. observed that classification-based approaches are among the most frequently used methods in financial fraud detection because they can learn from labelled transaction records and adapt better than conventional manual verification frameworks [1]. This observation is important for the present work because the proposed system also treats fraud detection as a supervised classification problem over historical transaction behaviour.

Dornadula and Geetha designed a credit card fraud detection approach that examined behavioural patterns through streaming transaction data and grouping strategies, showing that machine learning models can capture transaction regularities more effectively than static checks [2]. Their work reinforces the idea that fraud detection should be behaviour-centred rather than only amount-centred. In the UPI context, this idea remains relevant because fraud often depends on patterns involving type, timing, and transaction context.

Recent works also show strong interest in gradient boosting and ensemble-based models. Lingeswari and Brindha proposed an optimized online payment fraud framework using enhanced feature extraction and XGBoost-based classification, showing that boosting-based models can improve fraud identification in digital payment settings [4]. Similarly, Btoush and colleagues demonstrated that stacking and hybrid ensemble learning can improve predictive performance compared with single learners, especially in fraud datasets with complex structure [5]. Mia et al. further explored a hybrid fraud detection setting that integrated feature extraction, data balancing, and ensemble intelligence for high predictive power in imbalanced transaction datasets [6].

The present work shares the ensemble-learning motivation of these studies, but differs in several ways. First, it is framed specifically around a UPI-oriented fraud monitoring scenario rather than generic card fraud analysis. Second, the implemented system is not limited to offline classification; it includes an interactive dashboard that supports manual transaction input, visual risk scoring, simulation, and live monitoring. Third, the work incorporates explainability support through SHAP so that model decisions can be interpreted, which is important for user trust and analyst usability in financial applications [9]. Fourth, the system is designed as a modular pipeline with future API integration, which makes it more practical for extension into real-time service-oriented environments.

Another line of related work concerns the methodological foundations used in the present system. SMOTE is a classic and widely used approach for minority-class oversampling in imbalanced classification tasks [8]. XGBoost has become one of the most successful tree-boosting methods for structured prediction problems because of its efficiency and predictive strength [7]. SHAP, built on Shapley-value principles, has emerged as a consistent framework for interpreting model predictions at the feature level [9]. By combining these ideas within a UPI fraud detection pipeline, the current work connects fraud classification accuracy with transparency and monitoring usability.

Although existing literature offers strong methods for fraud detection, many studies focus either on algorithmic performance alone or on generalized transaction datasets. The gap addressed in this paper lies in combining ensemble

DOI: 10.17148/IJARCCE.2026.15419

fraud classification, explainable scoring, threshold-based decision support, and dashboard-oriented interaction in a single practical UPI monitoring workflow.

III. PROBLEM STATEMENT AND OBJECTIVES

A. Problem Statement

UPI-based digital payment systems process a very large number of transactions every day. Within this high-volume environment, fraudulent records form only a small minority, but their financial impact can be significant. Detecting such records is challenging because fraudulent behaviour may resemble legitimate user behaviour, especially when attackers imitate regular transaction patterns. Traditional systems based on predefined rules often fail to capture evolving fraud strategies, and manual review becomes inefficient as transaction volume grows.

The core problem addressed in this work is therefore the development of a data-driven fraud detection system capable of classifying transactions as legitimate or fraudulent with stronger adaptability, reduced dependence on static rules, and better support for analyst interpretation.

B. Objectives

The objectives of the proposed work are as follows:

1) To study the behaviour of financial transaction records relevant to UPI-oriented fraud detection.
2) To design a machine learning pipeline that can classify transactions into legitimate and fraudulent categories.
3) To compare a baseline Random Forest model with an improved XGBoost model for fraud detection.
4) To address severe class imbalance through suitable training-stage balancing techniques.
5) To produce transaction-level fraud probability scores rather than only hard class labels.
6) To improve interpretability using SHAP-based explanation of prediction drivers.
7) To build a Streamlit-based dashboard for manual prediction, risk visualization, live simulation, and monitoring analytics.

IV. PROPOSED METHODOLOGY

A. Overall System Design

The proposed system follows a modular pipeline. Transaction data is first collected from a structured CSV dataset, then cleaned and transformed into machine-usable form. Relevant features are selected and encoded. The processed data is used to train two ensemble models, namely Random Forest and XGBoost. The trained XGBoost model produces fraud probabilities for incoming transactions. A decision threshold is applied to convert probability into a fraud or safe label. The final result is exposed through an interactive dashboard and can also be integrated into an API-ready prediction flow.

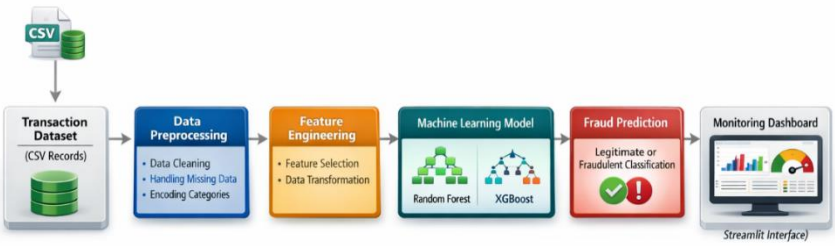

Fig. 1. System Architecture of the Proposed UPI Fraud Detection Framework.

B. Data Preparation and Preprocessing

The fraud detection workflow begins with a transaction dataset stored in CSV format. Based on the project implementation, important attributes include transaction step, transaction type, amount-related behaviour, balance-based attributes in the raw dataset, and the fraud label. Before model training, unnecessary columns are removed and categorical variables are transformed into numerical representations. This is required because machine learning algorithms such as Random Forest and XGBoost work best when categorical states are encoded into explicit numeric features.

Data preprocessing also ensures consistency between training-time transformation and inference-time transformation. This is important because a transaction submitted through the dashboard must pass through the same feature preparation logic used during training. Without this alignment, prediction reliability would degrade during deployment.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15413

C. Feature Engineering

A compact feature set was constructed to support fraud prediction. The implemented feature vector used in the final model consists of:

TABLE I: INPUT FEATURES USED IN THE FINAL FRAUD DETECTION MODEL

Feature Name	Description
step	Represents the transaction step or sequence in the dataset
fraud_rate_by_type	Indicates the historical fraud tendency associated with the transaction type
type_CASH_OUT	Binary encoded feature representing a cash-out transaction
type_DEBIT	Binary encoded feature representing a debit transaction
type_PAYMENT	Binary encoded feature representing a payment transaction
type_TRANSFER	Binary encoded feature representing a transfer transaction

This feature design is compact but meaningful. The temporal variable helps the model learn transaction progression patterns. The fraud-rate feature provides a historical risk indicator linked to transaction category. The one-hot encoded transaction-type features allow the model to distinguish behavioural differences across transaction modes. Such encoded features are appropriate for tree-based ensemble models because they preserve category-specific structure while remaining numerically usable.

D. Handling Class Imbalance

Class imbalance is one of the central challenges in fraud detection because the minority fraud class can be easily ignored by a classifier trained on overwhelmingly legitimate examples. To reduce this problem, SMOTE is incorporated into the pipeline so that the minority class is synthetically expanded during model preparation [3], [8]. This balancing strategy helps the model learn fraud patterns more effectively instead of being biased toward the dominant safe class.

The goal of balancing is not to distort the real-world class ratio, but to expose the classifier to enough minority examples during training so that suspicious behaviour is represented adequately in the learned decision boundary.

E. Model Development

Two ensemble models are used in the proposed system.

1) Random Forest: Random Forest is employed as the baseline model. It constructs multiple decision trees on random subsets of the training data and combines their outputs through voting. This model is robust, easy to interpret at a high level, and suitable for structured fraud data [2].

2) XGBoost: XGBoost is used as the improved model because of its ability to build trees sequentially, correct previous errors, and capture subtle non-linear patterns in structured data [4], [7]. Its efficiency and strong performance make it a suitable choice for the final fraud-scoring system.

The use of a baseline-plus-improved model strategy gives the work a comparative foundation. Rather than adopting a single method directly, the project first establishes reference behaviour and then improves performance using a stronger boosting-based classifier.

F. Probability-Based Fraud Scoring

Instead of relying only on binary output, the final model computes fraud probability. This is useful because fraud monitoring often benefits from risk-aware decision support rather than only yes/no classification. In the implemented system, the final classification rule is:

Fraud if $P(\text{fraud} \mid x) > 0.6$, otherwise Safe.

This threshold-based approach allows the model to be more conservative than a default 0.5 cutoff. In fraud detection, such threshold control is helpful because the cost of false alarms and missed frauds must be balanced according to application needs.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15413

G. Dashboard-Based Monitoring

The final stage of the methodology is visualization and monitoring. A Streamlit dashboard is used to support direct transaction input, probability display, risk-gauge output, feature-importance view, transaction simulation, and live monitoring table. This converts the model from a purely backend classifier into a usable monitoring prototype.

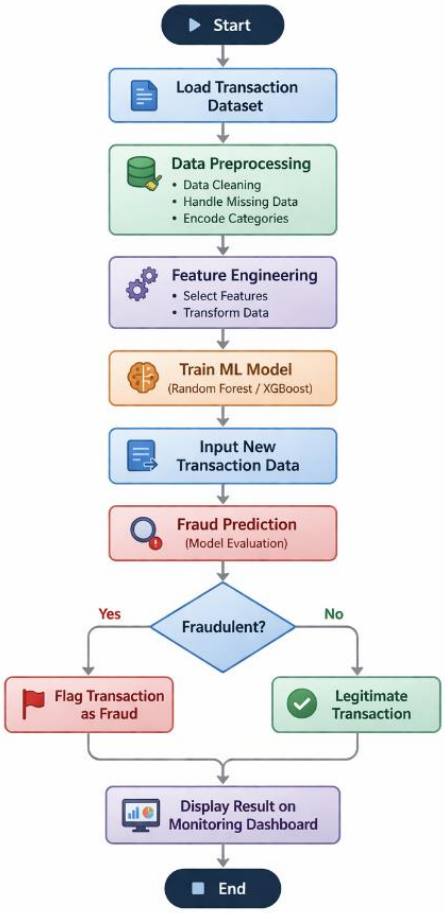

Fig. 2. Workflow of Fraud Prediction from Transaction Input to Final Dashboard Output.

V. IMPLEMENTATION DETAILS

A. Technology Stack

The system is implemented in Python using Pandas and NumPy for data handling, Scikit-learn and XGBoost for model development, Matplotlib/Seaborn/Plotly for visualization, Streamlit for dashboard development, and FastAPI for service integration support. This combination provides a practical software stack for fraud modelling and visualization.

B. Prediction Logic

The final implementation uses the XGBoost model saved as a serialized file and loaded into the dashboard at runtime. The model accepts the six-feature input vector described earlier. Transaction type is converted into one-hot indicators. Fraud probability is produced using predict_proba, and the decision threshold is set to 0.6. This creates a clear separation between scoring and classification.

The implementation supports the following transaction types:

1) PAYMENT
2) TRANSFER
3) CASH_OUT
4) DEBIT

A predefined fraud-rate map is associated with each type during manual dashboard entry. This allows controlled fraud-risk testing under category-aware conditions.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15413

C. Dashboard Functions

The dashboard includes multiple practical modules.

1) Manual prediction panel:

The user enters the transaction step and selects the transaction type. The system displays the fraud-rate value used for that type and generates the fraud prediction when the user presses the prediction button.

2) Fraud probability display:

The dashboard shows the predicted fraud probability in percentage form, which helps the user understand risk intensity rather than only the final class label.

3) Risk gauge:

A gauge chart represents fraud risk visually. Lower probability ranges indicate safer transactions, moderate ranges suggest caution, and high ranges indicate strong fraud suspicion.

4) Real-time transaction simulation:

The system can generate random synthetic transactions and evaluate them immediately. This helps demonstrate dynamic fraud scoring and system responsiveness.

5) Live fraud monitoring console:

Generated transactions are accumulated into a live table with transaction ID, type, step, fraud score, and status. Fraud and safe rows are highlighted for readability.

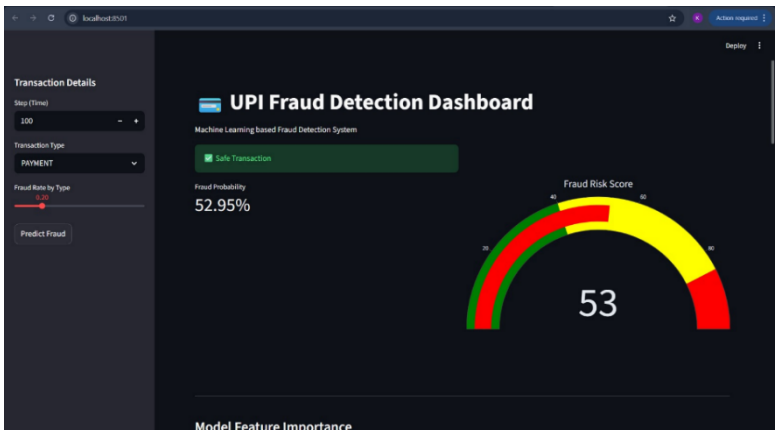

Fig. 3. Manual Fraud Prediction Interface with Risk Score Gauge.

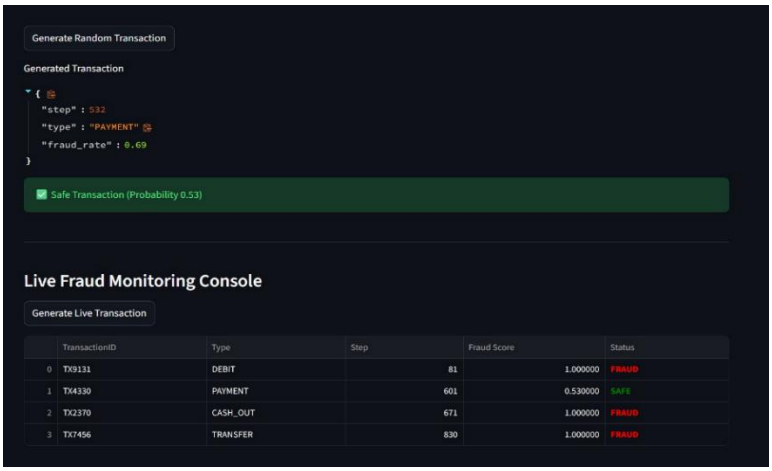

Fig. 4. Live Fraud Monitoring Console and Transaction Analytics.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15413

D. API Readiness

Although the main demonstration is built in Streamlit, the architecture also includes FastAPI support for service-oriented integration. This is useful because a practical fraud detection system may eventually receive transaction requests from external applications, payment gateways, or monitoring services. Designing the system with API compatibility in mind improves future extensibility.

VI. RESULTS AND ANALYSIS

A. Confusion Matrix-Based Evaluation

TABLE II: CONFUSION MATRIX OF THE FINAL FRAUD DETECTION MODEL

Actual Class	Predicted Safe	Predicted Fraud
Safe	314224	0
Fraud	181	168

This result shows that the model correctly classifies a very large number of legitimate transactions. The absence of false positives is especially notable because it indicates that normal transactions are not incorrectly blocked by the current decision threshold. In financial systems, false positives can reduce trust and inconvenience users, so this behaviour is practically meaningful. At the same time, the model misses 181 fraudulent transactions while correctly detecting 168 fraudulent transactions. This indicates that the model is conservative in its decision-making. It is extremely reliable when identifying safe transactions, but there is still room to improve minority-class recall.

From the confusion matrix, the following metrics can be derived:

Accuracy = (314224 + 168) / (314224 + 0 + 181 + 168) = 99.94%

Precision = 168 / (168 + 0) = 100.00%

Recall = 168 / (168 + 181) = 48.14%

F1-score = 2 × (1.00 × 0.4814) / (1.00 + 0.4814) ≈ 0.65

These derived values must be interpreted carefully. The accuracy is very high, but high accuracy is common in imbalanced datasets because the majority class dominates. Precision is perfect because the model produced no false positives in the recorded result. However, recall is moderate because a portion of actual fraud cases remains undetected. This means the current model is excellent for trustworthy positive fraud confirmation, but future enhancement should aim to improve fraud catch-rate without introducing too many false alarms.

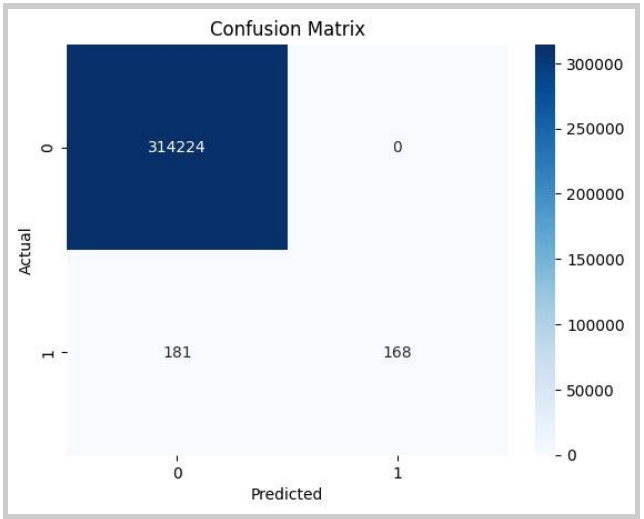

Fig. 5. Confusion Matrix of the Final XGBoost-Based Fraud Detection Model.

DOI: 10.17148/IJARCCE.2026.15413

B. Behavioural Interpretation of Results

The result profile suggests that the selected threshold of 0.6 makes the classifier strict before declaring a transaction as fraudulent. This explains the zero false-positive outcome. Such behaviour is often useful in user-facing payment systems because unnecessary fraud alerts can interrupt valid transactions and damage user confidence.

However, in analyst-driven back-office risk screening, a lower threshold might detect more suspicious cases at the cost of more false alarms. Therefore, threshold tuning remains an important future direction. Depending on deployment needs, the same model can be tuned either toward stronger fraud sensitivity or toward stricter safe-transaction confirmation.

C. Value of Dashboard Monitoring

The dashboard transforms the model from a research classifier into a monitoring prototype. Manual testing helps users understand how the model reacts to different transaction patterns. Live simulation demonstrates immediate scoring behaviour. The monitoring table and transaction analytics provide a visual environment that is closer to how fraud screening tools are used in practical settings.

The combined result shows that model performance and presentation layer together create a stronger system than a classifier alone. This is an important contribution of the present work.

VII. DISCUSSION

The proposed framework demonstrates that ensemble machine learning can provide an effective alternative to purely rule-based UPI fraud detection. Random Forest provides a stable baseline, while XGBoost improves the final classification process through stronger boosting-based learning. The use of SMOTE addresses minority-class underrepresentation during training, and SHAP improves interpretability after prediction. The dashboard adds operational visibility and user interaction.

One of the strengths of the system is its modular design. The same pipeline can be extended with additional features, new classifiers, deployment logic, or streaming transaction support. The prediction logic is compact, which makes it easier to maintain and explain. The decision threshold is explicitly controlled, which gives flexibility during deployment.

The main limitation visible from the current recorded results is fraud recall. Although the system is highly precise in identifying fraud and highly reliable in preserving legitimate transactions, a portion of fraudulent transactions remains undetected. This suggests that future improvements should focus on better fraud sensitivity through threshold tuning, richer behavioural features, sequential transaction context, graph-based relations, or hybrid anomaly-detection support [3], [5], [6].

VIII. CONCLUSION

This paper presented a hybrid machine learning framework for UPI fraud detection that combines baseline ensemble learning, improved boosting-based classification, explainable fraud scoring, and a real-time dashboard for monitoring. The proposed system was designed as a complete workflow rather than a single offline model. It includes dataset preparation, preprocessing, feature engineering, class imbalance handling, fraud probability estimation, Streamlit visualization, and API-oriented extensibility.

The implemented XGBoost-based model produced strong practical performance in the recorded evaluation. It achieved extremely high safe-transaction recognition and zero false positives in the documented confusion matrix, demonstrating strong reliability for legitimate transaction protection. The integration of explanation and monitoring features makes the system more useful than a plain classification script, because the fraud decision can be observed, interpreted, and demonstrated through an interactive interface.

IX. FUTURE SCOPE

Future work can extend the present system in several directions. Additional behavioural features such as transaction amount history, account-level activity windows, device information, and location-aware indicators may improve fraud recall. Threshold optimization can be explored to balance precision and recall according to operational needs. Real-time transaction streaming and deployment through secured APIs can make the system more production-ready. Graph-based modelling may help detect hidden relationships among suspicious accounts and transactions [5]. Incremental retraining

DOI: 10.17148/IJARCCE.2026.15413

can be incorporated so that the model adapts to new fraud patterns over time. Finally, the dashboard can be extended with alert prioritization, analyst notes, and case-management support.

REFERENCES

[1]. A. Ali, S. Shamsuddin, A. L. Ralescu, R. M. Musa, E.-S. M. El Houby, and H. I. Osman, "Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review," Applied Sciences, vol. 12, no. 19, p. 9637, 2022.

[2]. V. N. Dornadula and S. Geetha, "Credit Card Fraud Detection using Machine Learning Algorithms," Procedia Computer Science, vol. 165, pp. 631–641, 2019.

[3]. S. Kabane, "Impact of Sampling Techniques and Data Leakage on XGBoost Performance in Credit Card Fraud Detection," arXiv preprint arXiv:2412.07437, 2024.

[4]. R. Lingeswari and S. Brindha, "Online payments fraud prediction using optimized genetic algorithm based feature extraction and modified loss with XG boost algorithm for classification," Swarm and Evolutionary Computation, vol. 95, p. 101934, 2025.

[5]. E. A. L. M. Btoush, Y. Zhou, and others, "Enhancing credit card fraud detection with a stacking-based hybrid machine learning approach," PeerJ Computer Science, 2025.

[6]. Md. S. Mia, S. Roy, M. A. Ihsan, S. Hossain, and Md. K. U. Ahamed, "Data-driven financial fraud detection using hybrid artificial and quantum intelligence," BenchCouncil Transactions on Benchmarks, Standards and Evaluations, vol. 5, no. 4, p. 100252, 2025.

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

[8]. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[9]. S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4766–4777.

[10]. A. Al-Ghabban, H. M. Hussein, and A. T. Abdullah, "A Comprehensive Analysis of Credit Card Fraud Detection Using Hybrid Oversampling and Machine Learning Models," Al-Rafidain Journal of Engineering Sciences, vol. 4, no. 1, pp. 364–376, 2026.

[11]. T. Wongvorachan, S. He, and O. Bulut, "A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining," Information, vol. 14, no. 1, p. 54, 2023.

[12]. N. Damanik and C.-M. Liu, "Advanced Fraud Detection: Leveraging K-SMOTEENN and Stacking Ensemble to Tackle Data Imbalance and Extract Insights," IEEE Access, vol. 13, pp. 10356–10370, 2025.

