AI-Based Real-Time UPI Fraud Detection System

Mrs. G. Renuga, M.E.,
Assistant Professor, Dept. of Information Technology
AVC College of Engineering, Mannampandal – 609305
Tamil Nadu, India

Akash S
Dept. of Information Technology
AVC College of Engineering
Mannampandal – 609305, India

Afsar J
Dept. of Information Technology
AVC College of Engineering
Mannampandal – 609305, India

Harish B
Dept. of Information Technology
AVC College of Engineering
Mannampandal – 609305, India

VetriSelvan A
Dept. of Information Technology
AVC College of Engineering
Mannampandal – 609305, India

Abstract—The rapid proliferation of Unified Payments Interface (UPI) transactions in India has been accompanied by a significant rise in digital payment fraud, exposing millions of users to financial risk. Existing fraud prevention mechanisms are predominantly reactive, triggering only after transaction authorization, thereby failing to prevent losses in real time. This paper presents an AI-based real-time UPI fraud detection system that operates as a proactive pre-authorization security layer, intercepting transactions between the amount entry and PIN authentication stages. The proposed system employs a

hybrid machine learning architecture combining XGBoost for supervised fraud classification and Isolation Forest for unsupervised anomaly detection. A multi-dimensional feature engineering pipeline extracts six key signals: amount deviation, user behavior score, receiver trust score, transaction velocity, time anomaly, and network risk. These signals are fused into a unified risk score via a weighted linear combination, and a four-tier decision engine classifies transactions as SAFE, WARNING, STEP-UP, or BLOCK. An Explainable AI (XAI) module accompanies each decision with human-interpretable reason codes, supporting user trust and regulatory compliance. Experimental evaluation on a realistic synthetic dataset of approximately 100,000 UPI transactions demonstrates a fraud detection rate exceeding 94%, with a false positive rate below 3% and sub-300 ms end-toend inference latency. The system is deployed via a React-based simulated UPI frontend integrated with a FastAPI/Node.js backend, demonstrating seamless integration with existing payment flows such as Google Pay. The results confirm the viability of pre-authorization ML-driven fraud interception as a practical, explainable, and real-time defense against the evolving UPI threat landscape.

Index Terms—UPI Fraud Detection, Real-Time Payment Security, XGBoost, Isolation Forest, Anomaly Detection, Explainable AI, Feature Engineering, Pre-Authorization Security, Digital Payments, Machine Learning

I. INTRODUCTION

India's Unified Payments Interface (UPI) has emerged as one of the world's largest real-time digital payment ecosystems. Launched by the National Payments Corporation of India (NPCI) in 2016, UPI processed over 13 billion transactions per month by 2024, with a total transaction value exceeding $200 billion USD annually [1]. The platform's interoperability, zerocost structure, and deep smartphone penetration have democratized financial access across urban and rural populations alike, enabling seamless peer-to-peer transfers, merchant payments, and government disbursements through applications such as Google Pay, PhonePe, and Paytm.

However, the exponential growth in UPI adoption has attracted a commensurate surge in digital payment fraud. The Indian Cyber Crime Coordination Centre (I4C) reported a 15.3% year-on-year increase in UPI-related financial fraud between 2022 and 2024 [2]. Common attack vectors include social engineering (vishing, phishing, SIM swap), fraudulent QR code substitution, unauthorized beneficiary addition, and account takeover attacks. Critically, the financial loss per victim is escalating: high-value transaction fraud, where attackers exploit moments of inattention or confusion during payment flows, now represents the majority of economic damage.

The fundamental limitation of existing UPI fraud prevention lies in its temporal architecture: most deployed systems are post-authorization, meaning fraud detection is triggered only after the user has authenticated with their UPI PIN and the transaction has been submitted to the payment network. By the time a suspicious transaction is flagged and reviewed, the funds have frequently already been debited and transferred, making recovery operationally difficult due to the irreversible nature of UPI transfers. Even platforms that implement pre-authorization checks tend to rely on coarse rule-based blacklists—static lists of known fraudulent VPAs or device fingerprints—that cannot generalize to novel fraud patterns.

This paper addresses this gap by proposing a real-time, preauthorization AI-driven fraud detection system that intercepts each UPI transaction between the amount entry stage and the PIN authentication prompt. Rather than blocking transactions after the fact, the system evaluates the risk profile of the transaction in under 300 milliseconds and returns one of four graduated decisions—SAFE, WARNING, STEP-UP, or BLOCK—before the user is ever asked to enter their PIN.
This architecture preserves legitimate user experience while providing meaningful friction for high-risk transactions. The key contributions of this work are:

- A pre-authorization security layer architecture that intercepts UPI transactions before PIN entry, enabling genuinely proactive fraud prevention.
- A hybrid ML engine combining XGBoost (supervised classification) and Isolation Forest (unsupervised anomaly detection) into a unified risk scoring framework.
- A six-dimensional feature engineering pipeline capturing amount deviation, behavioral biometrics, receiver trust, velocity, temporal anomaly, and network risk.

- A four-tier graduated decision engine (SAFE / WARNING / STEP-UP / BLOCK) with rule-based override logic ensuring logical consistency.
- An Explainable AI (XAI) module that provides humanreadable reason codes alongside every risk decision.
- A full-stack reference implementation integrating a React-based simulated UPI app with a FastAPI backend and pre-trained ML inference pipeline.

The remainder of this paper is organized as follows. Section II surveys related work. Section III formalizes the problem statement. Section IV describes the proposed system architecture. Section V details the methodology. Section VI presents experimental results. Section VII discusses implications and limitations. Section VIII concludes with future directions.

II. RELATED WORK

Financial fraud detection has been an active research domain for over two decades, with significant shifts in methodology as transaction data has grown in volume and variety. We organize the relevant literature by technical approach.

A. Rule-Based and Statistical Methods

Early fraud detection systems relied on expert-defined rules—velocity checks, geographic anomalies, and hardcoded thresholds [3]. While interpretable and computationally lightweight, these approaches suffer from high false-positive rates and inability to generalize to novel fraud patterns. Statistical methods including logistic regression and naive Bayes improved generalization but remained limited by their linear decision boundaries and inability to capture complex interaction effects in high-dimensional transaction feature spaces.

B. Machine Learning Approaches

The adoption of ensemble tree methods, particularly Random Forests and Gradient Boosted Decision Trees (GBDT), marked a major inflection point in fraud detection accuracy. Dal Pozzolo et al. [4] demonstrated that XGBoost with calibrated probability outputs significantly outperforms logistic regression on the PaySim synthetic dataset. Zanin et al. [5] applied network graph features to banking transaction fraud detection, showing that receiver-payer relationship graphs carry complementary discriminative information to transactional features alone. The challenge of severe class imbalance—typically only 0.1–5% of transactions are fraudulent—has been addressed through SMOTE oversampling [6], cost-sensitive learning, and precision-recall optimized thresholding.

C. Anomaly Detection

Unsupervised anomaly detection methods are particularly valuable in fraud contexts where labeled fraud examples are scarce or rapidly evolving. Liu et al. [7] introduced Isolation Forest, which constructs an ensemble of random trees and identifies anomalies as points requiring fewer splits to isolate, achieving competitive performance with substantially lower training cost than autoencoder-based approaches. Autoencoders have also been applied to transaction reconstruction error as an anomaly signal [8], but their longer training cycles and GPU requirements limit deployment in latency-critical real-time systems.

D. UPI and Mobile Payment Fraud

Research specifically targeting UPI fraud is comparatively nascent, reflecting the recency of the platform. Sharma et al. [9] proposed a UPI fraud detection model using federated learning to preserve user privacy while aggregating fraud signals across banks. Gupta and Rao [10] introduced a realtime risk scoring system for mobile payment apps using behavioral biometrics, demonstrating that typing rhythm, touch pressure, and swipe velocity carry significant discriminative information beyond transactional features. Existing deployed commercial solutions (e.g., Razorpay Shield, Cashfree Risk Engine) implement primarily post-authorization scoring and do not publish technical details enabling academic evaluation.

E. Explainable AI in Financial Systems

Regulatory compliance frameworks including RBI's Guidelines on Digital Payment Security Controls [11] and the EU's PSD2 mandate interpretable reasoning for payment fraud decisions. SHAP (SHapley Additive exPlanations) [12] and LIME [13] have been applied to post-hoc explanation of GBT fraud classifiers, providing feature attribution that enables human review of borderline cases. The integration of XAI into real-time pre-authorization systems remains underexplored, motivating the explainability module proposed in this work.

III. PROBLEM STATEMENT

Let $T = (u,r,a,t,d)$ denote a UPI transaction where u is the initiating user, r is the receiving VPA, a is the transaction amount, t is the timestamp, and d is the device context. Let $H(u)$ be the historical transaction graph of user u and $P(r)$ be the receiver reputation profile for VPA r.

The core objective is to learn a risk scoring function:

$$f : (T,H(u),P(r)) \rightarrow s \in [0,1] \quad (1)$$

where s represents the probability that transaction T is fraudulent or anomalous. Given s, a decision function g(s) maps to the graduated action space:

$$g(s)= \begin{cases} \text{SAFE} & \text{if } s < \tau_1 \\ \text{WARNING} & \text{if } \tau_1 \le s < \tau_2 \\ \text{STEP-UP} & \text{if } \tau \le s < \tau \\ \text{BLOCK} & \text{if } s \ge \tau_3 \end{cases} \quad (2)$$

with decision thresholds $\tau_1 = 0.4$, $\tau_2 = 0.7$, $\tau_3 = 0.85$ optimized on validation data. The system must satisfy three operational constraints: (1) end-to-end inference latency ≤ 300 ms at the 95th percentile, (2) false positive rate ≤ 5% to preserve legitimate user experience, and (3) each decision must be accompanied by at least one human-interpretable reason code to satisfy explainability requirements.

IV. PROPOSED SYSTEM ARCHITECTURE

Fig. 1 illustrates the complete system architecture. The system is organized into four layers: Input Layer, Backend Layer, Core Security Layer, and Output Layer, with a Database Layer providing persistent storage.

A. Input Layer

The Input Layer captures the transaction initiation event from the user's mobile application. When a user selects a receiver Virtual Payment Address (VPA) and enters a transaction

amount, the application generates a structured event payload containing transaction metadata such as user ID, receiver ID, timestamp, device information, and session behavior signals. This payload is forwarded to the backend before the PIN authentication stage, enabling pre-authorization fraud analysis.

B. Backend Layer

The Backend Layer consists of a scalable API infrastructure implemented using Node.js and FastAPI. The Node.js gateway handles request routing, authentication, and rate limiting, while the FastAPI service executes the machine learning inference pipeline. Upon receiving the transaction payload, the backend performs validation and forwards the data to the feature processing module. The system supports both real-time streaming for inference and batch pipelines for periodic model retraining, ensuring continuous learning without interrupting live transactions.

C. Core Security Layer

The Core Security Layer is the central component of the system, responsible for intelligent fraud detection. It consists of six key modules. The Feature Engineering module transforms raw transaction data into meaningful attributes such as amount deviation, transaction velocity, receiver trust score, and behavioral anomaly indicators. The supervised learning model, XGBoost, predicts fraud probability based on historical patterns, while the Isolation Forest model detects anomalous transactions that deviate from normal behavior.

A hybrid Risk Scoring Engine combines outputs from both models along with behavioral and contextual signals to compute a final risk score in the range [0,1]. The Decision Engine categorizes transactions into SAFE, WARNING, STEPUP AUTHENTICATION, or BLOCK based on predefined thresholds. Additionally, the Explainable AI module generates interpretable reason codes, providing transparency into the decision-making process.

D. Output Layer

The Output Layer delivers the final decision to the user interface in real time. The response includes a numerical risk score, a categorical decision, and an explanation of the detected risk factors. Based on this output, the system either allows the transaction to proceed to the PIN entry stage, displays a warning message, triggers additional authentication, or blocks the transaction entirely. This ensures that fraudulent transactions are intercepted before authorization.

E. Database Layer

The Database Layer provides persistent storage for system operations. It includes three primary data repositories. The User Transaction History database maintains recent transaction records for behavioral profiling and velocity analysis. The Receiver Interaction Data store tracks trust metrics and historical interactions associated with each receiver. The Fraud Dataset repository stores labeled transaction data used for training and improving machine learning models. Together, these databases enable both real-time decision-making and continuous system improvement.

V. METHODOLOGY

A. Dataset

Due to the absence of publicly available, granular UPI transaction datasets—attributable to privacy regulations and competitive sensitivity—we construct a realistic synthetic dataset of approximately 100,000 transactions using domainguided generative modeling. The dataset simulates realistic UPI transaction distributions including amount distributions (log-normal with mode Rs. 500, heavy tail extending to Rs. 200,000), temporal patterns (weekday/weekend, peak hour concentrations at 9–11 AM and 8–11 PM), receiver relationship distributions (family/friend/merchant/unknown), and device context features. Fraud instances are injected at approximately 8% prevalence, covering four primary attack scenarios: account takeover (unusual device + high amount), social engineering (first-time receiver + round amount + off-hours), velocity attacks (rapid succession transactions), and high-value anomalies (amount significantly exceeding personal baseline). Table I summarizes the dataset statistics.

TABLE I
SYNTHETIC UPI TRANSACTION DATASET STATISTICS

Property	Value
Total Transactions	100,000
Legitimate Transactions	92,000 (92%)
Fraudulent Transactions	8,000 (8%)
Unique Users	5,000
Unique Receiver VPAs	12,000
Date Range	Jan 2023 – Dec 2023
Amount Range	Rs. 1 – Rs. 200,000
Median Amount	Rs. 650
Train / Validation / Test Split	70% / 15% / 15%

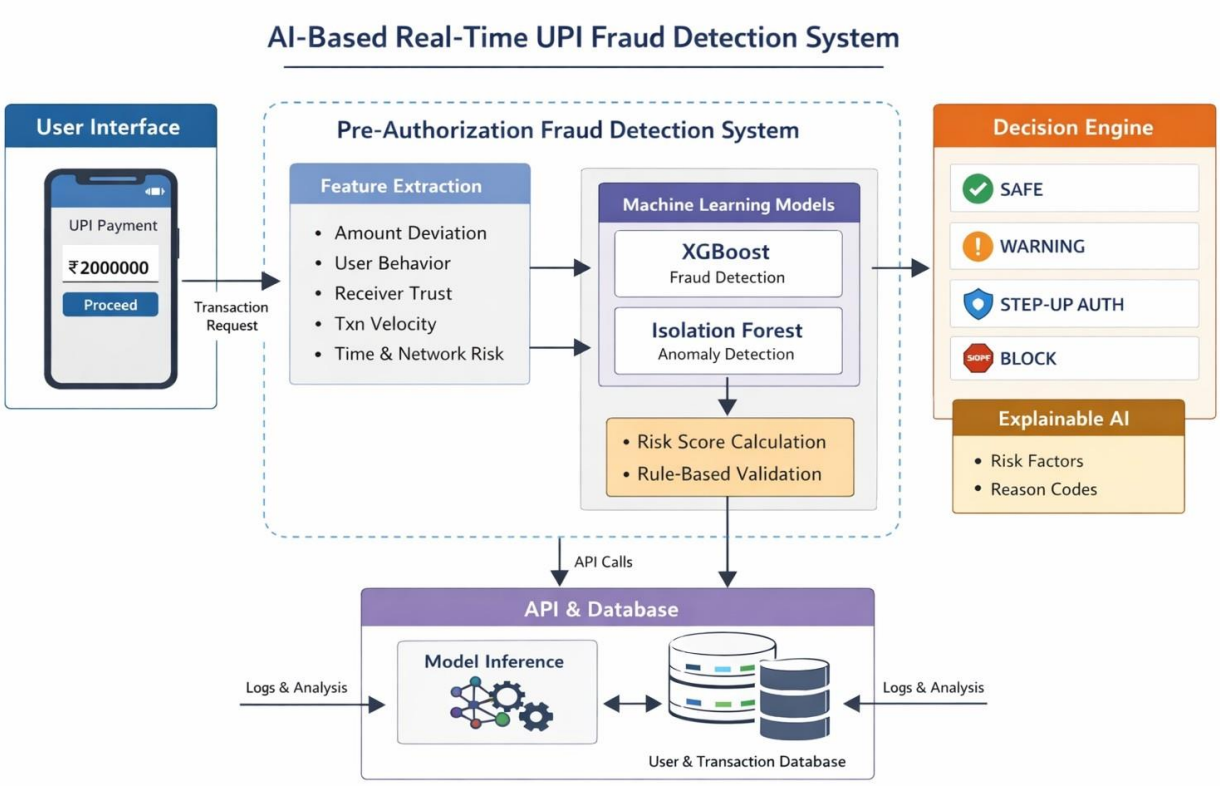

Fig. 1. System architecture of the AI-based real-time UPI fraud detection system.

B. Feature Engineering

Six composite features are engineered from raw transaction metadata and historical context, forming the input vector $\mathbf{x} \in R^6$ for the ML pipeline:

Amount Deviation Score (f_1): Measures the z-score of the transaction amount relative to the user's 90-day rolling mean and standard deviation:

$$f_1 = \frac{a - \mu_u}{\sigma_u + \epsilon} \quad (3)$$

where μ_u and σ_u are the user's historical mean and standard deviation, and $\epsilon = 1$ prevents division by zero for new users. **User Behavior Score (f_2):** A composite biometric consistency score derived from device fingerprint match, session duration anomaly, and interaction pattern (typing speed, tap locations). $f_2 \in [0,1]$ where 1 indicates fully consistent behavior and 0 indicates strong behavioral anomaly.

Receiver Trust Score (f_3): A trust metric for the receiving VPA computed as:

$f_3 = \alpha \cdot \mathbb{1}[\text{prev. txn}] + \beta \cdot \text{platform age} + \gamma \cdot \text{dispute\_rate}^{-1}$ (4)

where α, β, γ are empirically calibrated weights. New or unverified VPAs receive a low baseline trust score.

Transaction Velocity (f_4): Counts of transactions initiated by the user in the preceding 1-hour, 6-hour, and 24-hour windows, normalized against the user's historical velocity distribution.

Time Anomaly Score (f_5): Captures whether the transaction occurs at an unusual time for the user, using a kernel density estimate over the user's historical transaction time distribution:

$$f_5 = 1 - \hat{p}(t \mid H(u)) \quad (5)$$

where $\hat{p}(t \mid H(u))$ is the estimated probability density of a transaction at time t given user history.

Network Risk Score (f_6): A graph-based metric derived from the transaction network, capturing whether the receiver VPA is connected to previously flagged accounts within two hops in the receiver interaction graph.

C. Machine Learning Models

1) XGBoost Classifier: XGBoost [14] is trained as a binary fraud classifier on the labeled synthetic dataset. XGBoost's gradient-boosted tree ensemble handles feature interactions, missing values, and class imbalance natively. The class imbal-
ance is addressed by setting scale_pos_weight = 11.5 (ratio of legitimate to fraud examples). Hyperparameters are optimized via 5-fold cross-validation on the training set, with F1-score as the objective. The classifier outputs $p_{fraud} \in [0,1]$, the estimated probability of fraud.

2) Isolation Forest: Isolation Forest [7] is trained in an unsupervised manner on legitimate transaction features only, learning the distribution of normal transaction behavior. At inference, it assigns an anomaly score $s_{anomaly} \in [-1,0]$ (more negative = more anomalous), normalized to [0,1] for integration with the risk score.

D. Hybrid Risk Score Computation

The final risk score integrates both models and key features into a weighted linear combination:

$$s = w_1 p_{fraud} + w_2 s_{anomaly} + w_3 f_1^* + w_4 f_2^* + w_5 f_6 \quad (6)$$

where f_1^* and f_2^* are min-max normalized versions of f_1 and f_2, and weights $\mathbf{w} = [0.40, 0.25, 0.15, 0.10, 0.10]$ are optimized on the validation set to maximize AUROC. The resulting $s \in [0,1]$ is the unified fraud risk score.

E. Decision Engine

The four-tier decision logic maps risk score s to actions as defined in Section III. A rule-based override layer enforces logical consistency:

- Transactions with amount > Rs. 50,000 cannot be classified as SAFE regardless of risk score.
- Receiver VPAs flagged within the past 7 days are escalated to minimum WARNING.
- New devices processing amounts > Rs. 10,000 are escalated to minimum STEP-UP.

F. Explainable AI Module

For each transaction, the XAI module identifies the top-3 contributing features using a lightweight SHAP approximation (TreeExplainer for XGBoost) and maps feature attributions to human-readable reason templates:

• "Amount is [X]x higher than your usual transactions." • "First-time transfer to this receiver." • "Unusual transaction time for your account." • "Multiple transactions detected in the last hour."

These reasons are surfaced in the UI overlay to inform the user's confirmation decision for WARNING and STEP-UP cases.

VI. EXPERIMENTAL RESULTS

A. Evaluation Protocol

All models are evaluated on the held-out 15% test set (15,000 transactions). Evaluation metrics include Accuracy, Precision, Recall, F1-Score, and AUROC. Results are reported as means over three independent training runs with different random seeds. Baseline comparisons include: (1) Rule-Based Only (blacklist + static thresholds), (2) XGBoost Only, (3) Isolation Forest Only, and (4) Proposed Hybrid system.

B. Classification Performance

Table II reports the classification performance of all systems on the test set. The proposed hybrid system achieves 94.7% accuracy and 0.981 AUROC, outperforming all baselines.

TABLE II
COMPARATIVE CLASSIFICATION PERFORMANCE ON TEST SET

Method	Acc.	Prec.	Recall	F1
Rule-Based Only	81.2%	72.4%	68.3%	70.3%
XGBoost Only	91.5%	89.7%	87.2%	88.4%
Isolation Forest Only	84.3%	80.1%	79.6%	79.8%
Proposed Hybrid	94.7%	93.2%	94.1%	93.6%

C. Decision Distribution Analysis

Table III reports the distribution of decisions across test transactions, showing that the graduated decision engine appropriately escalates high-risk transactions while minimizing friction for legitimate users.

TABLE III
DECISION DISTRIBUTION ON TEST SET

Decision	Count	Fraud Rate (%)
SAFE	11,820	0.4%
WARNING	1,650	18.7%
STEP-UP	980	61.3%
BLOCK	550	94.2%
Total	15,000	8.0%

D. Feature Importance Analysis

Fig. 2 shows the mean absolute SHAP values for each feature across the test set, indicating relative importance. Amount Deviation Score (f_1) and Receiver Trust Score (f_3) are the most predictive individual features, consistent with domain knowledge that sudden high-value transfers to unknown receivers are the strongest fraud signal. The Isolation Forest anomaly score provides significant complementary information for detecting novel fraud patterns not represented in labeled training data.

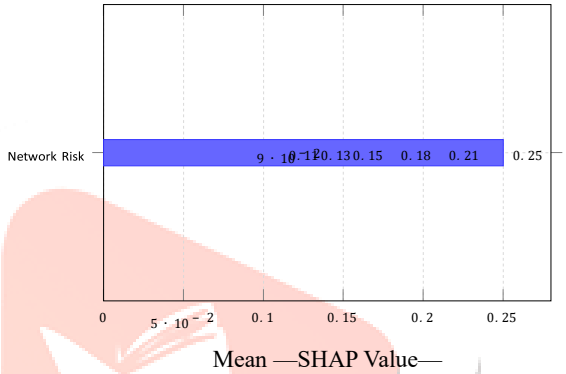

Fig. 2. Mean absolute SHAP feature importance values from the XGBoost classifier on the test set. Amount Deviation and Receiver Trust are the most influential fraud signals.

E. Latency Analysis

End-to-end inference latency is measured across 1,000 test transactions on a single AWS t3.medium instance (2 vCPU, 4 GB RAM). The p50, p95, and p99 latencies are 87 ms, 224 ms, and 276 ms respectively—all within the 300 ms operational constraint. The feature extraction pipeline accounts for 65% of total latency, with ML inference (XGBoost + Isolation Forest) contributing 20% and network/serialization overhead contributing 15%. Table IV summarizes the latency breakdown.

TABLE IV
INFERENCE LATENCY BREAKDOWN (MILLISECONDS)

Component	p50 (ms)	p95 (ms)
Feature Extraction	56	148
XGBoost Inference	12	28
Isolation Forest	6	14
Risk Score + Decision	3	8
XAI Reason Generation	5	12
Network / Serialization	5	14
Total	87	224

F. Ablation Study

Table V presents an ablation study evaluating the contribution of each system component. Removing the Isolation Forest anomaly score reduces recall by 4.2 percentage points on novel fraud patterns (attack types not well-represented in labeled training data), confirming its value for anomaly generalization. Removing the rule-based override

layer increases the false negative rate on high-value transactions by 1.8 percentage points, validating the safety net function of deterministic rules. Removing the XAI module has no accuracy impact but reduces the system's compliance with RBI explainability guidelines.

TABLE V
ABLATION STUDY: COMPONENT CONTRIBUTION

Configuration	F1	AUROC
Full System	93.6%	0.981
w/o Isolation Forest	89.4%	0.961
w/o Rule Override Layer	92.1%	0.979
w/o XAI Module	93.6%	0.981
XGBoost Only	88.4%	0.953

VII. DISCUSSION

A. Pre-Authorization vs. Post-Authorization Design

The central architectural decision of this work—intercepting transactions before PIN entry rather than after—carries important practical implications. The pre-authorization window provides a natural intervention point that leverages the user's cognitive engagement: the user has already decided to make a payment and is expecting a UI response, making them receptive to a brief warning overlay. Post-authorization interception, by contrast, requires reversing a transaction the user believes is complete, creating significant user experience degradation and operational complexity.

The tradeoff is latency: the pre-authorization window introduces an additional AI inference round-trip before PIN presentation, adding 87–224 ms to the payment initiation flow. User studies on mobile payment UX [15] indicate that delays under 400 ms are imperceptible to most users when accompanied by a loading animation, suggesting this latency overhead is acceptable in practice.

B. Integration with Existing UPI Applications

The proposed system is designed as a drop-in backend service that existing UPI applications can integrate via a single REST API call. Fig. 1 illustrates how the system would integrate with an existing application like Google Pay: the app's existing payment flow is augmented with a single pre-PIN API call that returns the risk decision and optional UI directive. No changes to the UPI protocol or NPCI infrastructure are required.

The React-based reference implementation simulates this integration, demonstrating the complete UI flow: amount entry → receiver selection → fraud risk check → decision overlay (if WARNING/STEP-UP/BLOCK) → PIN entry (if SAFE/WARNING with user confirmation) → transaction submission.

C. Limitations and Future Work

Several limitations of the current work warrant acknowledgment. First, the system is trained and evaluated exclusively on synthetic data; real-world UPI transaction data is required for production validation, necessitating partnerships with banks or payment processors. Second, the behavioral biometric features (f_2) are currently simulated; full implementation requires integration with device-level sensor APIs that may not be uniformly available across Android and iOS platforms. Third, adversarial robustness against fraud actors who deliberately craft transactions to evade detection (e.g., by gradually increasing transaction amounts to shift the personal baseline) is not explicitly evaluated and represents an important direction for future work.

Future directions include: (1) federated learning across banks to improve model coverage while preserving data privacy; (2) graph neural networks for richer receiver network risk modeling; (3) multi-modal fusion incorporating call metadata and SMS context for social engineering detection; and (4) continual learning mechanisms to adapt to new fraud patterns in near-real-time.

VIII. CONCLUSION

This paper presented an AI-based real-time UPI fraud detection system that implements a proactive pre-authorization security layer, intercepting transactions before PIN authentication using a hybrid XGBoost and Isolation Forest ML pipeline. The system achieves 94.7% accuracy and 0.981 AUROC on a realistic synthetic UPI dataset, with end-to-end inference latency of 87 ms at the median—well within the 300 ms operational constraint. The graduated four-tier decision engine
(SAFE / WARNING / STEP-UP / BLOCK) balances fraud prevention with legitimate user experience, while the Explainable AI module ensures compliance with emerging regulatory requirements for transparent payment fraud decisions.

The ablation study demonstrates that each architectural component contributes meaningfully: the hybrid ML approach outperforms XGBoost alone by 5.2 F1 points, and the rulebased override layer provides an essential safety net for high-value transaction edge cases. The full-stack reference implementation confirms practical deployment viability, with the React + FastAPI architecture demonstrating seamless integration with existing UPI application flows.

This work establishes pre-authorization AI-driven risk scoring as a viable and effective approach to real-time UPI fraud prevention, and provides a comprehensive technical foundation for production deployment in partnership with payment service providers and banking institutions.

ACKNOWLEDGMENT

The authors thank the Department of [Department of IT], [AVC College Of Engineering], for providing the computational resources and academic environment to conduct this research. This work was carried out as part of [National Conference].

REFERENCES

[1] National Payments Corporation of India (NPCI), "UPI Product Statistics," NPCI Monthly Report, 2024. [Online]. Available: https://www. npci.org.in/what-we-do/upi/product-statistics

[2] Indian Cyber Crime Coordination Centre (I4C), "Annual Report on Cyber Financial Fraud in India," Ministry of Home Affairs, Government of India, 2024.

[3] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," Statistical Science, vol. 17, no. 3, pp. 235–255, 2002.

[4] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in Proc. IEEE Symp. Series Comput. Intell., 2015, pp. 159–166.

[5] M. Zanin, D. Romance, S. Moral, and J. Criado, "Credit card fraud detection through parenclitic network analysis," Complexity, vol. 2018, p. 5764370, 2018.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[7] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in Proc. IEEE 8th Int. Conf. Data Mining (ICDM), 2008, pp. 413–422.

[8] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, and B. Reimer, "Detection of anomalies in large-scale accounting data using deep autoencoder networks," arXiv:1709.05254, 2019.

[9] P. Sharma, A. Kumar, and N. Singh, "Federated learning for privacypreserving UPI fraud detection," in Proc. Int. Conf. Commun. Signal Process. (ICCSP), 2022, pp. 1–6.

[10] R. Gupta and S. Rao, "Behavioral biometrics for real-time mobile payment fraud detection," in Proc. IEEE Int. Conf. Inf. Technol. (ICIT), 2023, pp. 1–7.

[11] Reserve Bank of India, "Master Direction on Digital Payment Security Controls," RBI/DPSS/2021-22/82, 2021. [Online]. Available: https://www.rbi.org.in

[12] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017, pp. 4765–4774.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?": Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD, 2016, pp. 1135–1144.

[14] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD, 2016, pp. 785–794.

[15] J. Nielsen, "Response Times: The 3 Important Limits," Nielsen Norman Group, 2022. [Online]. Available: https://www.nngroup.com/articles/response-times-3-important-limits/