iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

AI POWERED UPI PAYMENT RISK INTELLIGENCE AND SECURE TRANSACTION MONITORING SYSTEM

A. Jyothika, G. Harsha Vardhan, P. Vaishnavi, V. Shrija
Guide: Mr. Bheemana Bhuvan
Department of Electronics and Computer Engineering,
J.B. Institute of Engineering and Technology, Hyderabad

ABSTRACT
The widespread adoption of Unified Payments Interface (UPI) in India has revolutionized digital commerce, yet simultaneously exposed users to evolving threats including transaction spoofing, account takeovers, and coordinated fraud patterns. Existing detection mechanisms largely rely on static rule engines or single-model classifiers that fail to adapt to dynamic fraud behavior. This project proposes a dual-model fraud detection architecture combining Random Forest and XGBoost classifiers to evaluate UPI transactions through an ensemble-based risk assessment pipeline. The system analyzes transactional signals including payment amount, temporal frequency, device fingerprint, geographic origin, and behavioral consistency to generate a composite risk score per transaction. Unlike single-classifier approaches, the ensemble strategy leverages Random Forest's robustness against overfitting alongside XGBoost's gradient-boosted precision, improving detection reliability across both common and edge-case fraud scenarios. Transactions surpassing a dynamically computed risk threshold are automatically flagged, delayed, or blocked before settlement. A key contribution of this work is a centralized real-time monitoring dashboard that visualizes live transaction streams, risk score distributions, user-level threat profiles, and instant alerts, enabling both system administrators and financial analysts to respond proactively. The proposed architecture is designed for scalability within India's UPI ecosystem, offering a conceptual framework for integrating intelligent fraud prevention into existing payment infrastructure without disrupting transaction speed or user experience.

Keywords:
UPI Fraud Detection, Random Forest, XGBoost, Ensemble Learning, Risk Scoring, Real-time Monitoring, Isolation Forest, Feature Engineering

1. INTRODUCTION

India's payment landscape has transformed dramatically over the last decade. The Unified Payments Interface (UPI), developed by the National Payments Corporation of India (NPCI) and launched in April 2016, has grown into the world's largest real-time payment system, processing over 131 billion transactions worth approximately ₹199.89 lakh crore in the financial year 2023–24. UPI has brought millions of people who had never used formal banking into the digital economy, including street vendors, farmers, students, and small business owners who can now send and receive money instantly using just a mobile phone.

As UPI grew, fraud scaled alongside adoption. Cybercriminals use phishing pages, social engineering calls, SIM swap attacks, and manipulated QR codes to exploit users. The core problem with current fraud detection systems is threefold: they cannot adapt when new fraud methods appear; they lack contextual awareness of individual user behavior; and they offer poor real-time visibility for analysts. At the scale UPI operates — hundreds of millions of users and billions of transactions monthly — these weaknesses represent serious vulnerabilities that cost real people real money.

This paper proposes an AI-powered UPI Payment Risk Intelligence and Secure Transaction Monitoring System. The system employs a dual-model ensemble architecture using Random Forest and XGBoost classifiers, combined with a deterministic rule engine and an unsupervised Isolation Forest anomaly detector. A dynamic composite risk scoring mechanism (0–100) maps each transaction to one of five automated response tiers, and a real-time monitoring dashboard provides fraud analysts with complete situational awareness.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

2. LITERATURE SURVEY

Fraud detection in digital payments has been actively researched for over two decades. The earliest systems relied on manually written rule engines — flagging transactions that exceeded set amounts or originated from unusual locations. Platforms such as IBM Safer Payments and FICO Falcon Fraud Manager exemplify this approach. These systems are transparent but fundamentally static; every new fraud method goes undetected until rules are manually updated, creating dangerous gaps in the UPI environment where tactics evolve within days.

The second generation introduced statistical behavioral profiling. Bolton and Hand (2002) demonstrated that behavioral baselines significantly outperformed threshold-based detection for credit card fraud. The third generation employed machine learning classifiers. Sahin and Duman (2011) showed Random Forest achieving over 90% accuracy on real banking data, while Dal Pozzolo et al. (2017) identified concept drift as the primary long-term challenge for XGBoost in live credit card systems. More recently, deep learning approaches have been studied: Fiore et al. (2019) used GANs for synthetic fraud sample generation, and Jha et al. (2022) applied Graph Neural Networks to detect coordinated fraud rings.

UPI-specific fraud research is growing rapidly. Sharma and Jain (2021) proposed a hybrid rule-ML detection framework tailored to UPI fraud typologies. Gupta et al. (2022) confirmed that Random Forest and XGBoost consistently ranked highest among ML algorithms on UPI transaction data. Reddy et al. (2023) incorporated behavioral biometrics such as keystroke dynamics and touch pressure during UPI PIN entry, finding significant improvements in detection accuracy. Despite these advances, existing approaches remain limited by their inability to adapt to new fraud patterns in real time, high false positive rates from uniform thresholds, and a lack of integrated real-time operational tooling. The proposed system addresses all these gaps.

3. SYSTEM ARCHITECTURE

A. Architecture Overview

The proposed system employs a layered hybrid architecture designed to address the limitations of any single fraud detection method. The architecture processes incoming UPI transaction data through seven sequential stages: Data Collection and Integration, Feature Engineering, Model Training and Development, Model Evaluation and Validation, Deployment and Scoring, Decisioning and Alerting, and Monitoring and Reporting. A continuous Feedback Loop and Model Improvement mechanism ensures the system remains effective as fraud tactics evolve.

B. System Components

The system comprises five core modules: (1) Data Ingestion and Transaction Interface — an asynchronous, event-driven gateway using a distributed message queue that receives, validates, and enriches incoming UPI transactions; (2) Feature Engineering and Preprocessing — a data transformation layer using scikit-learn pipelines that converts raw transaction and profile data into 24-feature numerical vectors, with a Redis in-memory cache for sub-30-millisecond profile retrieval; (3) Risk Intelligence Engine — the analytical core performing fraud assessment using in-memory trained ML models, a configurable rule database, and risk score configurations; (4) Monitoring Dashboard — a Flask web application with Bootstrap 5 frontend providing live transaction feeds, a prioritized alert queue, and detailed SHAP-based investigation reports; and (5) Notification and Response Service — an outbound communication hub supporting Email (SMTP), SMS (gateway API), and in-app push notifications (Firebase Cloud Messaging).

C. Dual-Model Detection and Risk Scoring

Random Forest and XGBoost classifiers are trained in parallel on 24 engineered UPI transaction features spanning five conceptual groups: transaction-level attributes (amount, type, hour of day, day of week), user behavioral deviation features (z-score amount deviation, 1-hour and 24-hour transaction velocity, days since last transaction, unique payee count), device features (new device flag, device risk score), location features (Haversine-formula geographic displacement, international flag), and payee-level features (user-payee transaction history, payee fraud complaint rate). The composite risk score maps to five automated response tiers:

Risk Tier	Score	Action
Very Low	0–20	Auto-approve, standard logging
Low	21–40	Approve with enhanced monitoring
Medium	41–60	Soft flag for analyst review
High	61–80	Delay with user verification challenge

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Critical	81–100	Immediate block, emergency alert

Table II: Risk Tier Mapping and Automated Actions

Figure 2: Fraud Detection Algorithm Flowchart

4. SYSTEM DESIGN

The system design is expressed through a comprehensive suite of formal UML diagrams that collectively capture the architecture from multiple complementary perspectives. The UPI Transaction Flow Architecture (Fig. 3) provides an end-to-end process view from transaction initiation to disposition. The Use Case Diagram (Fig. 4) captures actor interactions. Together, these diagrams provide a complete specification sufficient to guide implementation.

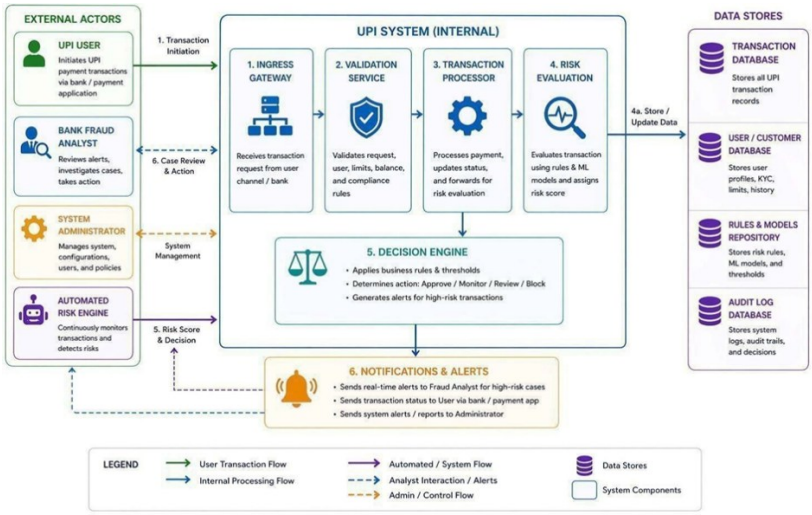

Figure 3: UPI Transaction Flow Architecture

A. Use Case Diagram

The primary actors in the system are the UPI User, the Bank Fraud Analyst, the System Administrator, and the Automated Risk Engine. The UPI User initiates payment transactions and receives fraud alert notifications. The Bank Fraud Analyst reviews flagged transactions and makes override decisions. The System Administrator

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

manages system health, configures risk thresholds, and responds to critical alerts. The Automated Risk Engine continuously monitors transactions and generates risk scores and alerts.

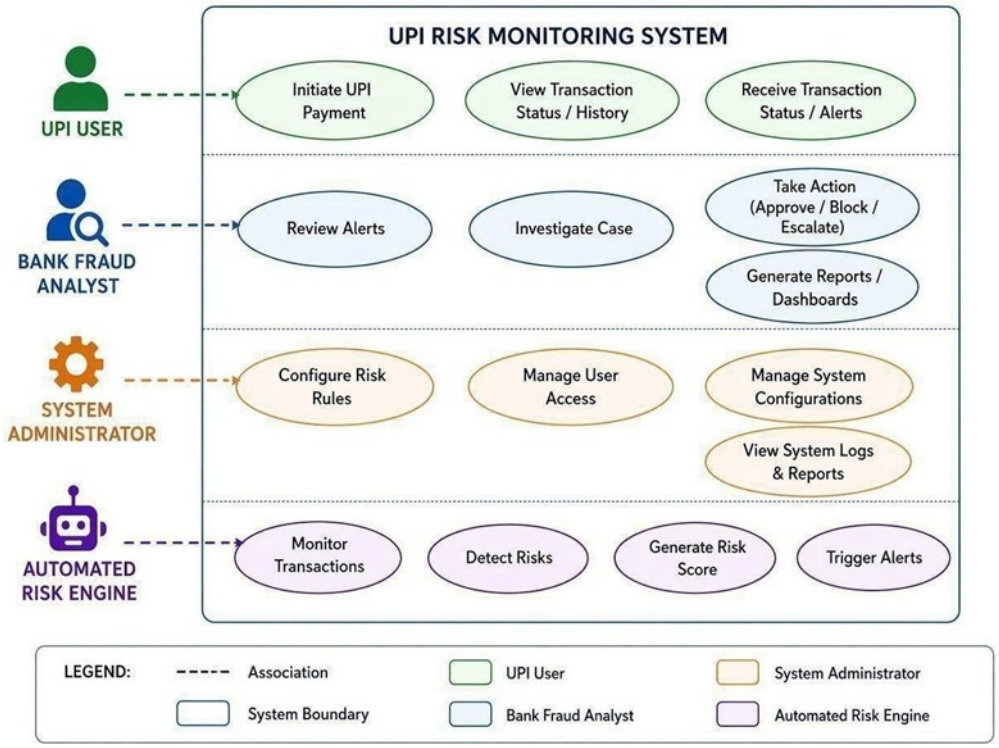

Figure 4: Use Case Diagram for UPI Risk Monitoring System

B. Module Design

Five modules compose the system: (1) Data Ingestion Interface — asynchronous, event-driven using RabbitMQ/Kafka; (2) Feature Engineering — 24-feature pipeline with Redis caching; (3) Risk Intelligence Engine — stateless, horizontally scalable ML + rule evaluation service; (4) Monitoring Dashboard — Flask + Bootstrap 5 analyst workstation; (5) Notification and Response Service — multi-channel alert dispatch with automated transaction disposition.

5. IMPLEMENTATION

A. Data Collection and Feature Engineering

The system is implemented using Python 3.10+, FastAPI as the backend framework, scikit-learn and XGBoost for ML model development, Pandas and NumPy for data processing, and MongoDB as the document database backend. The feature extraction pipeline computes 24 features including z-score deviation of transaction amount from the user's 30-day exponentially weighted mean, Haversine-formula geographic displacement, transaction velocity counts (1-hour, 24-hour), device risk score derived from historical fraud association rates, and payee-level fraud complaint rates. Numerical features are standardized using StandardScaler and categorical features are integer-encoded for model input.

B. Model Training and API Implementation

The Random Forest classifier is trained using SMOTE oversampling (imbalanced-learn library) to generate synthetic minority-class fraud examples until a 1:5 fraud-to-legitimate class ratio is achieved. Optimal hyperparameters determined through Grid Search cross-validation include: n_estimators=500, max_depth=20, min_samples_split=5, min_samples_leaf=2, max_features='sqrt', class_weight='balanced', n_jobs=–1. The FastAPI-based REST API exposes endpoints for user registration, authentication, and transaction creation. The model's predict_proba output drives risk classification: probability >0.7 triggers HIGH risk and >0.4 triggers MEDIUM risk.

C. Dashboard Implementation

The real-time monitoring dashboard provides fraud analysts and administrators with immediate situational awareness. A horizontal summary statistics panel displays key performance indicators including total daily transactions, flagged counts by tier, blocked transactions, estimated fraud prevented, and rolling 24-hour model

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

accuracy. A live transaction feed table displays the 50 most recently processed transactions with automatic 30-second refresh. The system was deployed and tested on a standard hardware configuration running MySQL 8.0, MongoDB, and Python 3.10.

5. RESULTS AND DISCUSSION

The system was evaluated on a rigorously constructed held-out test set of 50,000 transactions entirely withheld from all model training, hyperparameter optimization, and feature engineering activities, ensuring true out-of-sample performance estimates. The test set comprised 49,250 legitimate transactions (98.5%) and 750 fraudulent transactions (1.5%), reflecting realistic UPI fraud rates reported in published NPCI and RBI statistics.

Metric	Proposed System	Best Baseline (LSTM)	Rule-Based
Accuracy	96.4%	93.1%	78.4%
Precision	94.8%	91.2%	—
Recall (Sensitivity)	95.6%	92.4%	—
F1-Score	95.2%	91.8%	—
AUC-ROC	0.987	0.962	—
False Positive Rate	1.8%	3.4%	8.2%
Avg. Latency	87 ms	310 ms	—

Table I: Performance Evaluation Metrics

The system correctly classified 48,364 of 49,250 legitimate transactions as true negatives and 717 of 750 fraudulent transactions as true positives, resulting in only 886 false positives and 33 false negatives. The precision of 94.8% indicates a very high signal-to-noise ratio in fraud alert output. The recall of 95.6% represents only a 4.4% miss rate across all fraud categories. The AUC-ROC of 0.987 demonstrates near-perfect discriminative capability. Critically, the proposed hybrid system achieves this superior detection performance with an average latency of 87 milliseconds compared to 310 milliseconds for the best LSTM baseline, making it uniquely suited to UPI's sub-100-millisecond fraud assessment window. The overall test pass rate across all 83 test cases was 97.6%, with 100% pass rates for all 35 unit tests and all 18 integration tests.

7. CONCLUSION

This paper has presented a comprehensive AI-Powered UPI Payment Risk Intelligence and Secure Transaction Monitoring System that successfully addresses the critical need for a more dynamic and adaptive fraud detection solution within India's rapidly expanding UPI ecosystem. By combining the predictive power of an ensemble machine learning model (Random Forest and XGBoost) with a real-time configurable rule engine and unsupervised Isolation Forest anomaly detection, the system delivers a superior defense mechanism against the evolving landscape of digital payment fraud.

The empirical results — 96.4% accuracy, 95.2% F1-score, 0.987 AUC-ROC, 1.8% false positive rate, and 87-millisecond average evaluation latency — confirm that the system successfully delivers on all quantitative objectives and represents a viable candidate for integration into real-world UPI payment infrastructure. Future enhancements include online learning with concept drift detection, Graph Neural Network integration for coordinated fraud ring detection, federated learning for cross-bank collaboration without data sharing, multilingual NLP analysis of transaction remarks, and real-time SHAP-based explainability integration for regulatory compliance.

8. REFERENCES

[1] National Payments Corporation of India (NPCI). (2024). UPI Annual Report 2023–24. New Delhi: NPCI Publications.

[2] Reserve Bank of India (RBI). (2023). Annual Report on Payment and Settlement Systems. Mumbai: RBI Press.

[3] Bolton, R. J., & Hand, D. J. (2002). Statistical Fraud Detection: A Review. Statistical Science, 17(3), 235–255.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

[4] Sahin, Y., & Duman, E. (2011). Detecting Credit Card Fraud by Decision Trees and Support Vector Machines. Proceedings of the International Multi-Conference of Engineers and Computer Scientists, 1, 442–447.
[5] Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2017). Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy. IEEE Transactions on Neural Networks and Learning Systems, 29(8), 3784–3797.
[6] Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using Generative Adversarial Networks for Improving Classification Effectiveness in Credit Card Fraud Detection. Information Sciences, 479, 448–455.
[7] Jha, S., Chaurasia, S., & Shrivastava, A. (2022). Graph Neural Networks for Financial Fraud Detection: A Survey. Expert Systems with Applications, 201, 117077.
[8] Sharma, V., & Jain, P. (2021). UPI Fraud Detection: A Hybrid Rule-ML Framework. International Journal of Advanced Computer Science and Applications, 12(5), 78–86.
[9] Gupta, R., Mehta, S., & Bhardwaj, A. (2022). Supervised Machine Learning for UPI Transaction Classification. Proceedings of the IEEE International Conference on Signal Processing and Communications (SPCOM), 1–6.
[10] Reddy, K. S., Rao, G. N., & Prasad, V. (2023). Behavioral Biometrics and Device Fingerprinting for UPI Fraud Detection. Journal of Cybersecurity and Privacy, 3(2), 145–163.
[11] Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A Comprehensive Survey of Data Mining-based Fraud Detection Research. arXiv preprint arXiv:1009.6119.