

Paysure: Smart UPI Payments with Real-Time Fraud Detection.

Saniya Sudhir Raut¹, Sushant Shahaji Gaikwad², Nidhi Yugesh Sansare³, Priti Vikram Pawar⁴

Dept. of Information Technology, Sinhgad Institute of Technology & Science (SPPU), Pune, India

Abstract - PaySure — Smart UPI Payment with Real-Time Fraud Detection is a full-stack web application designed to address the growing threat of financial fraud within India's Unified Payments Interface ecosystem. As UPI has expanded to process billions of transactions every month, the speed and openness that define the platform have simultaneously made it a high-value target for fraud attacks. Existing prevention mechanisms — static rule engines and post-transaction review systems — prove structurally inadequate in this context: rule engines are circumvented by fraudsters who adapt to their fixed conditions, and post-transaction detection arrives only after the money has already left the victim's account. PaySure closes these gaps by embedding an intelligent fraud evaluation engine directly into the transaction initiation flow, ensuring that every payment is assessed before any funds are transferred. The system combines a deterministic rule-based scoring engine with a CatBoost gradient boosting classifier trained on real UPI transaction data, fusing their output into a unified risk score that produces one of four outcomes — GREEN, YELLOW, RED, or BLOCKED — while a receiver UPI fraud database provides a deterministic override that guarantees confirmed fraudulent accounts are always refused.

The platform serves three user roles — regular account holders, merchants, and administrators — each with dedicated workflows and role-enforced access controls. Merchants receive a dynamically generated UPI QR code and a received-payments log; administrators have full audit visibility across all users and transactions. SHAP — SHapley Additive explanations — is integrated into the detection pipeline to surface the specific features driving every fraud decision in human-readable language, making outcomes transparent and auditable. An IMEI-based device registry automatically blacklists devices associated with repeated fraudulent activity. The system achieves a classification accuracy of 97% and an AUC of 0.9967, and is deployed to Render.com with a

PostgreSQL backend and environment-configurable fraud thresholds.

Key Words: UPI fraud detection, real-time fraud prevention, CatBoost gradient boosting, hybrid fraud scoring, SHAP explainability, Flask web application, rule-based fraud engine, IMEI device tracking, explainable artificial intelligence, transaction risk assessment, digital payment security, cloud deployment, fraud risk scoring, anomaly detection in payments.

1. INTRODUCTION

The Unified Payments Interface (UPI), introduced by the National Payments Corporation of India (NPCI) in 2016, has revolutionized digital payments by enabling interoperable and instant bank transactions through mobile applications. UPI has become one of the most widely used payment systems in India due to its simplicity, speed, and accessibility. The platform processes billions of transactions every month and serves as a critical pillar of India's digital economy.

Despite its widespread success, the rapid expansion of UPI has significantly increased exposure to cyber-enabled financial frauds. Fraudsters exploit vulnerabilities through phishing attacks, fake UPI handles, QR code tampering, social engineering, SIM swap attacks, and unauthorized payment requests. Traditional fraud prevention systems rely heavily on static rules and manual verification mechanisms, which are insufficient against adaptive fraud strategies.

Most existing fraud detection systems operate reactively, identifying fraudulent activity only after transactions are completed. Such delayed detection often results in irreversible financial losses and diminished user trust. Therefore, there is an urgent need for proactive and intelligent fraud detection systems capable of identifying suspicious activity in real time before payment authorization.

To address this challenge, this paper proposes **PaySure**, a machine learning-based real-time UPI fraud detection system.

The framework integrates a high-performance CatBoost gradient boosting classifier within a Flask-based web application, selected after comparative evaluation against other machine learning models. The proposed solution performs pre-transaction fraud assessment by analysing transaction patterns, user behavior, and network-related features to generate instant fraud alerts.

The major contributions of this work are:

- I. Development of a real-time fraud detection framework for UPI transactions.
- II. Comparative evaluation of CatBoost, XGBoost, and Random Forest models.
- III. Application of SMOTE-based balancing to address class imbalance.
- IV. Implementation of SHAP explainability for transparent fraud prediction.
- V. Deployment of the trained model using a Flask-based web application for low-latency fraud assessment.

The proposed framework aims to improve transaction security, enhance user trust, and provide scalable AI-driven fraud prevention for digital payment systems.

2. LITERATURE REVIEW

The rapid growth of Unified Payments Interface (UPI) transactions has increased the risk of digital payment fraud, creating a need for intelligent fraud detection systems. Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), and anomaly detection techniques have been widely explored to identify fraudulent UPI transactions and improve transaction security.

Anjali et al. [1] developed a machine learning-based framework using Logistic Regression, Random Forest, and Support Vector Machine models to detect fraudulent UPI transactions through behavioral and transactional features. Kumar et al. [2] proposed an AI-driven mechanism for detecting both UPI fraud and QR code tampering using ML and DL techniques. Rao et al. [3] compared Random Forest, XGBoost, and CatBoost classifiers and demonstrated that ensemble learning approaches provide superior precision and recall compared to single classifiers. Rethisha and Cyindia [4] focused on low-latency real-time fraud detection using Logistic Regression, Decision Tree, and Random Forest models, emphasizing computational efficiency

for high-volume transaction screening. Trinity and Sharma [5] analysed common UPI fraud techniques, including phishing and impersonation attacks, while discussing existing rule-based prevention strategies and highlighting the need for AI/ML integration.

Several researchers have also investigated advanced learning approaches for fraud detection. Kumar et al. [6] utilized deep learning techniques to model sequential transaction behavior and identify anomalies in UPI payments. Bharath et al. [7] introduced the Hybrid Markov Learning Methodology (HMLM), which detects fraud by identifying deviations from normal transaction sequences. Gupta et al. [8] employed Recurrent Neural Networks (RNNs) to leverage the sequential and time-dependent nature of UPI transaction records for fraud classification. Tamilselvi et al. [9] experimentally evaluated an elevated deep learning methodology and demonstrated improved detection performance using sophisticated neural network architectures. Rani et al. [10] developed the SecureUPI system based on the Random Forest algorithm and emphasized the importance of feature engineering for improving fraud detection accuracy.

Recent studies have further explored privacy-preserving and anomaly-based detection methods. Devassy et al. [11] proposed FedUPI, a Federated Learning framework that enables collaborative model training among banks without sharing sensitive user data. K. S. A. et al. [12] employed the Isolation Forest algorithm for unsupervised anomaly detection and demonstrated its effectiveness in identifying rare fraudulent transactions within highly imbalanced datasets.

Although existing fraud detection systems achieve promising accuracy, many approaches focus on specific detection techniques such as machine learning, deep learning, federated learning, or anomaly detection individually. There remains a need for an integrated fraud detection framework that combines multiple detection strategies while providing efficient real-time transaction assessment.

To address this challenge, PaySure combines a rule-based fraud engine with a CatBoost machine learning classifier to perform real-time fraud assessment before transaction completion. The system integrates SHAP-based explainability, receiver UPI fraud database verification, device-level monitoring, and multi-

level risk classification to provide an accurate, transparent, and scalable fraud prevention framework for UPI transactions.

3. PROBLEM DEFINITION

The central problem PaySure addresses is that existing UPI fraud prevention mechanisms are too slow, too rigid, or too opaque to protect users effectively. Post-transaction detection acts after the money has already moved; static rule engines are circumvented the moment a fraudster learns their fixed thresholds and cannot learn from new patterns without manual intervention. No single transaction attribute — amount, time, category, or device — is sufficient to identify fraud on its own, because risk emerges from the combination of multiple signals evaluated together. Fraud also takes several distinct forms simultaneously — known-fraudster UPI numbers, compromised devices, newly created accounts, and frequency-based split-transaction attacks — each demanding a different technical response that no single conventional mechanism can provide. Finally, automated financial decisions must be explainable; a blocked transaction with no stated reason erodes user trust and leaves administrators unable to audit or defend the system's behaviour.

Objectives

To systematically address the problem above, the following objectives were defined for PaySure:

1. Build a full UPI payment simulation platform with secure registration, dual-mode login, and a personalised dashboard showing live fraud statistics.
2. Implement a multi-rule fraud engine covering amount limits, daily caps, hourly frequency, suspicious hours, new account risk, high-risk categories, and device blacklisting.
3. Train a CatBoost classifier on 2,666 real UPI transaction records across ten features, achieving at least 95% accuracy and AUC of 0.95.
4. Combine rule and ML scores (60% ML, 40% rules) into a unified risk value mapped to GREEN, YELLOW, RED, or BLOCKED, with a database override that always blocks confirmed fraudsters.

5. Integrate SHAP explainability to surface the top risk-contributing features as human-readable reasons on every transaction result page.
6. Provide a full administrator panel for user management, role promotion, transaction audit across all users, and merchant oversight.
7. Deploy to Render.com with PostgreSQL, environment-variable-driven fraud thresholds, and a public health-check endpoint.

4. PROPOSED METHODOLOGY

The proposed system, PaySure, employs a structured machine learning (ML) pipeline for real-time UPI fraud detection. The methodology is divided into seven major phases — Data Collection, Data Understanding, Exploratory Data Analysis (EDA), Data Pre-processing and Feature Engineering, Model Training and Ensemble Learning, Threshold Tuning, and Feature Explainability — each contributing to building a robust and explainable fraud detection framework.

A. Data Collection:

A synthetic dataset simulating UPI transactions was created, containing attributes such as transaction time, amount, user demographics, transaction category, location, and fraud labels. An additional feature, `ip_address`, was generated to simulate network-based behavioral patterns. This dataset formed the foundation for training and evaluating fraud detection models.

B. Data Understanding and Exploration:

An initial assessment of the dataset structure, quality, and consistency was performed to identify missing values, duplicates, and data imbalances. Using Python libraries like *Pandas* and *NumPy*, the dataset was examined through descriptive statistics (`df.info()`, `df.describe()`) and data distribution checks. The analysis revealed a mild class imbalance between fraudulent and non-fraudulent transactions, guiding further pre-processing and sampling strategies.

C. Exploratory Data Analysis (EDA):

EDA was conducted to uncover relationships and trends within the data. Visualizations such as bar plots, boxplots, and heatmaps were generated using *Matplotlib* and *Seaborn* to

compare fraud vs. non-fraud transaction distributions, transaction amounts, and correlations among numerical features. Insights indicated that higher transaction amounts, certain categories, and specific transaction times exhibited a stronger likelihood of fraud. Repetitive IP addresses also showed significant correlation with fraudulent activity, highlighting the relevance of behavioral and network-level features.

D. Data Pre-processing and Feature Engineering:

This phase ensured data cleanliness and model readiness. Missing values were handled through removal or imputation, while outliers were treated using the Interquartile Range (IQR) method. Categorical variables such as transaction category, state, and IP address were encoded numerically using Label Encoding, and continuous variables were normalized using StandardScaler. The dataset was split into 80% training and 20% testing sets using stratified sampling to maintain class balance. Additionally, Synthetic Minority Oversampling Technique (SMOTE) was applied to handle class imbalance and enhance model generalization.

E. Model Training and Ensemble Learning:

Multiple machine learning models were trained, including Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Support Vector Machine (SVM), XGBoost, and CatBoost. Model performance was evaluated using Accuracy, Precision, Recall, F1-score, ROC-AUC, and PR-AUC metrics. Among all, CatBoost, XGBoost, and Random Forest demonstrated superior performance in fraud classification. While a Soft Voting Ensemble combining CatBoost and Random Forest was implemented and evaluated during the development phase, the standalone CatBoost classifier was selected for production deployment because it exhibited superior individual predictive performance (97.46% F1-score vs. 97.13% for the ensemble) and lower runtime inference latency.

F. Threshold Tuning:

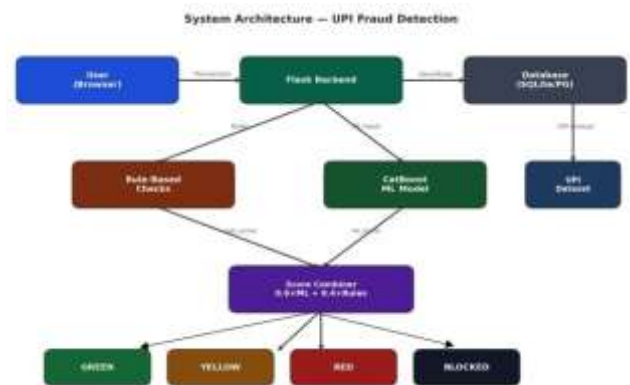
Threshold tuning was employed to balance the trade-off between Precision (reducing false positives) and Recall (capturing actual frauds). Instead of using the default 0.5

probability cut-off, multiple thresholds were tested. The mathematically optimal prediction threshold was identified at 0.4081 based on the maximum F1-score, achieving an empirical Precision of 97.7% and a Recall of 97.1% on the test set. This calibration ensures that the model effectively identifies fraudulent transactions while minimizing unnecessary alerts for legitimate users.

G. Model Explainability using SHAP Analysis:

To ensure transparency and trust, SHAP (SHapley Additive exPlanations) analysis was applied to interpret feature importance and individual predictions. SHAP results revealed that transaction amount, transaction category, and time were the most influential predictors of fraud likelihood. This interpretability confirms that model decisions align with real-world fraud patterns, ensuring both accuracy and explainability — essential for financial applications.

System Architecture:



The proposed PaySure system adopts a multi-layered, modular architecture designed to perform real-time fraud detection prior to the completion of a UPI transaction. The architecture is composed of six interconnected components: the Client Interface, the Flask Backend, the Database, the Rule-Based Checks module, the CatBoost ML Model, and the Score Combiner, each contributing a distinct functional role in the fraud assessment pipeline.

The process is initiated when a user submits a UPI transaction request through the web-based client interface. This request, containing transaction metadata such as the recipient UPI ID, transaction amount, and device information, is transmitted to the Flask Backend, which serves as the central processing unit of the system. The backend orchestrates all downstream

operations, simultaneously dispatching the transaction data to the rule-based and machine learning modules while maintaining a persistent connection with the underlying database for record storage and retrieval.

The database layer, implemented using SQLite or PostgreSQL, is responsible for storing user credentials, transaction history, and device mapping records. In parallel, it performs a UPI lookup against the stored UPI dataset to determine whether the recipient UPI ID has any prior association with fraudulent activity. This historical context enriches the fraud assessment by incorporating longitudinal behavioral patterns into the decision-making process.

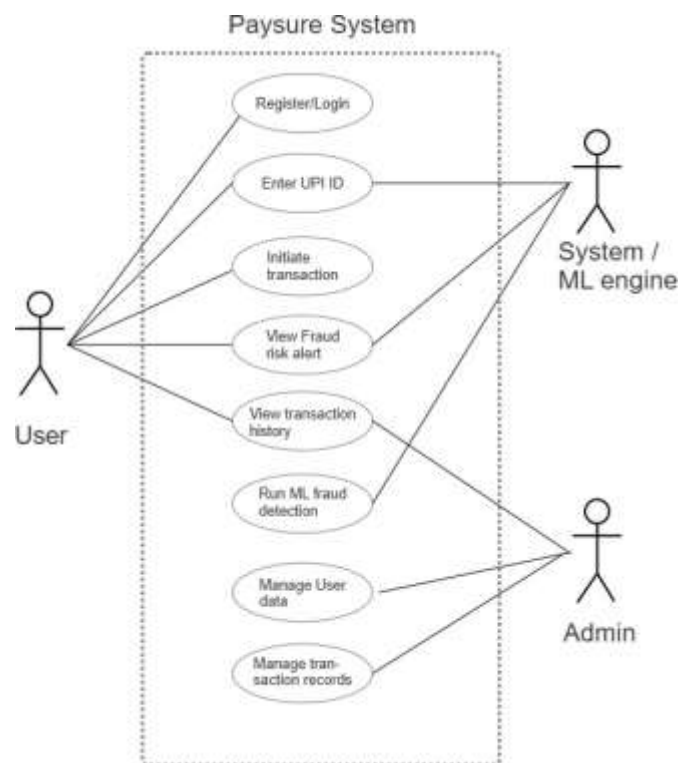
The Rule-Based Checks module evaluates the transaction against a set of predefined security heuristics, including transaction velocity thresholds, blacklisted UPI handles, and anomalous amount patterns. This deterministic component generates a quantitative rule score reflecting the degree to which the transaction violates established fraud indicators. Concurrently, the CatBoost ML Model, trained on an extensively pre-processed dataset using SMOTE-balanced class distributions, analyses the behavioral and demographic features of the transaction to compute a probabilistic fraud score, referred to as the ml score.

Both scores are subsequently forwarded to the Score Combiner, which produces a unified fraud risk score using a weighted aggregation formula: $\text{Final Score} = 0.6 \times \text{ml_score} + 0.4 \times \text{rule_score}$. The machine learning component is assigned a higher weight of 60% in recognition of its superior predictive accuracy, while the rule-based component contributes the remaining 40%, ensuring that deterministic domain knowledge is retained in the final assessment.

Based on the resulting composite score, the system classifies the transaction into one of four risk categories. A GREEN classification indicates a low-risk transaction that is permitted to proceed without interruption. A YELLOW classification signals a moderate risk level, prompting a cautionary alert to the user while still allowing the transaction to continue. A RED classification denotes a high-risk transaction, triggering a strong warning that advises the user against proceeding. A BLOCKED classification is assigned to transactions exhibiting critical fraud indicators, resulting in the automatic termination of the payment process before any funds are transferred.

This layered architectural design ensures that fraud detection is both comprehensive and computationally efficient. By combining the interpretability of rule-based logic with the predictive power of ensemble machine learning, PaySure delivers accurate, transparent, and low-latency fraud assessments that intervene proactively at the pre-transaction stage, thereby safeguarding users from financial loss in real time.

Use Case Diagram:



The use case diagram for the PaySure system illustrates the functional interactions between three primary actors — the User, the System/ML Engine, and the Admin — and the core use cases within the PaySure system boundary.

The User actor represents the end-user interacting with the PaySure web interface. The user initiates interaction by completing the Register/Login process for authenticated access. Upon login, the user enters a recipient UPI ID and Initiates a Transaction. The system then returns a fraud risk alert classifying the transaction as GREEN, YELLOW, RED, or BLOCKED, enabling an informed decision before payment. The user can also View Transaction History to review past payment activities and their associated risk outcomes.

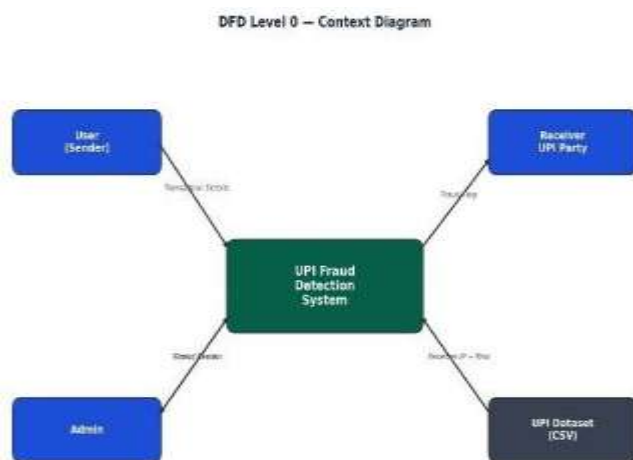
The System/ML Engine actor represents the automated backend intelligence of PaySure. It processes the entered UPI ID through a dataset lookup, executes the ML Fraud Detection

using the trained standalone CatBoost classifier, and delivers the final weighted risk classification to the user interface in real time.

The Admin actor is responsible for maintaining the platform's data integrity. The admin manages User Data and Transaction Records within the database, and can View Transaction History to monitor system-wide activity, identify recurring fraud patterns, and support compliance requirements.

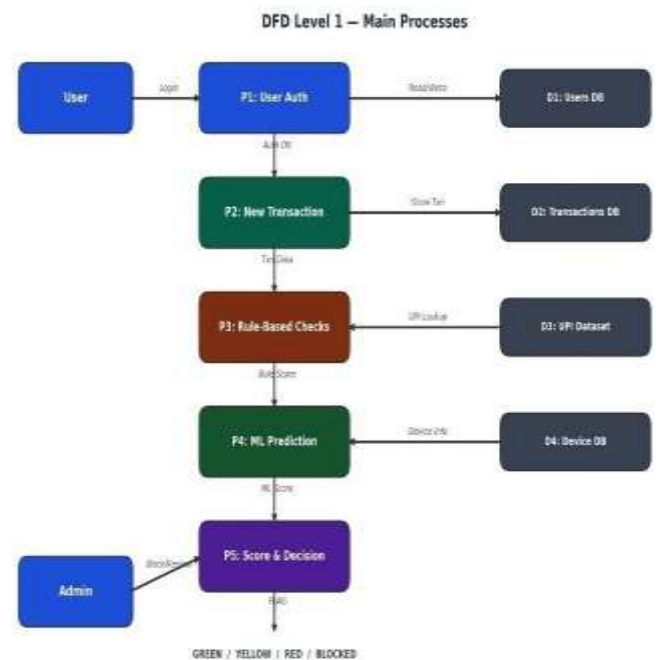
Together, these three actors and their associated use cases define the complete functional scope of the PaySure system, demonstrating that fraud detection is an integrated, actor-driven workflow in which the user, the ML backend, and the administrative layer each play a well-defined role in ensuring secure and reliable UPI transactions.

DFD Level 0:



The DFD Level 0 Context Diagram provides a high-level overview of the PaySure UPI Fraud Detection System and its interactions with four external entities. The User (Sender) submits transaction details to the central system, which processes the fraud assessment and returns a fraud result to the Admin for monitoring and review. Simultaneously, the system issues a fraud flag to the Receiver UPI Party to notify them of the transaction's risk status, while the UPI Dataset (CSV) supplies receiver IP and risk reference data to support the detection process. This diagram establishes the system's boundaries and defines the primary data flows between all external actors and the core fraud detection engine.

DFD Level 1:



The DFD Level 1 diagram decomposes the PaySure system into five core internal processes that together execute the complete fraud detection workflow. The flow begins when the User submits login credentials to P1: User Authentication, which reads and writes user records from the D1: Users DB and issues an Auth OK signal upon successful verification. The authenticated session proceeds to P2: New Transaction, where transaction data is stored in D2: Transactions DB and forwarded for analysis. P3: Rule-Based Checks consults the D3: UPI Dataset via a UPI lookup and validates device status in the D4: Device DB to compute a rule score. Concurrently, P4: ML Prediction processes the transaction features using the trained CatBoost model to compute an. Both scores converge at P5: Score and Decision, where the Admin can intervene via a Block/Review action, and the system produces a final FLAG output classifying the transaction as GREEN, YELLOW, RED, or BLOCKED.

5. IMPLEMENTATION

The PaySure system was implemented using Python and the Flask framework, integrating a trained CatBoost classifier for real-time UPI fraud detection. The implementation consists of three core components — the machine learning model, the Flask API backend, and a web-based user interface.

The ML model was trained on preprocessed transaction data and serialized using *Joblib* for deployment. The Flask-based

backend handles user inputs (UPI ID, transaction amount, device/IP details), processes them through the model, and returns a fraud probability score. Based on a tuned threshold, the system instantly classifies the transaction as *High Risk* or *Safe* and displays an alert to the user.

A responsive web interface was built using HTML, CSS, and Flask templates to simulate a UPI-style environment. It includes:

- Registration Page: For new users to sign up securely.



The registration page is titled "Create an Account". It features a form with the following fields: First Name, Last Name, Username, Email, Mobile Number (with a "No sign mobile" link), Date of Birth (with a "dd-mm-yyyy" format), Full Address (with a "Full address (optional)" label), State (a dropdown menu), ZIP Code, Gender (a dropdown menu), and Password (with a "Confirm Password" field). A green "Register" button is at the bottom. A link "Already have an account? Login" is also present.

- Login Page: For authenticated access to the dashboard.



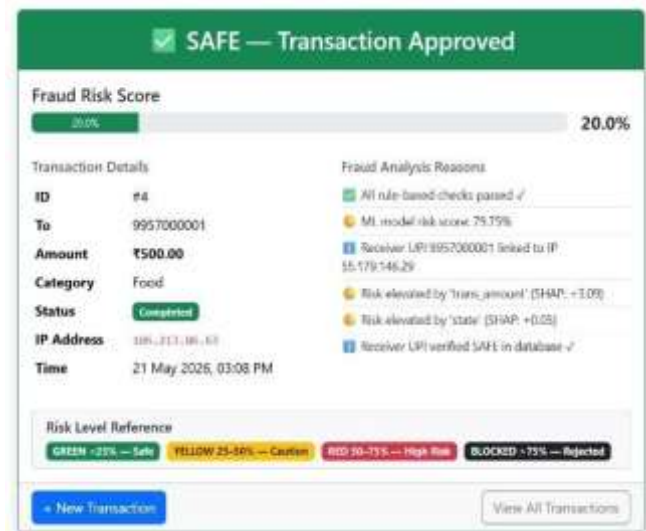
The login page is titled "Login". It has a form with fields for "Email/Username" and "Password". Below the fields is a "Remember Me" checkbox and a blue "Login" button. There are links for "Not with an account?" and "Forgot your password?". A "Use Case: Demo Login" button is also visible.

- Dashboard/Home Page: Displays fraud detection options, transaction history, and API health.



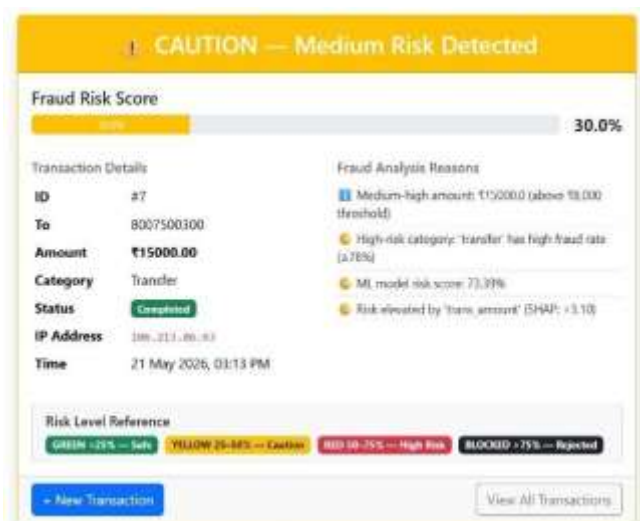
The dashboard is titled "Dashboard" and shows a summary of the system. It includes a "UPI ID" field, a "Total Transactions" counter (9), a "Completed" counter (6), and a "Blocked Fraud" counter (2). There are also links for "Your Profile", "Make a Payment", and "Transaction History".

- Safe Transaction:



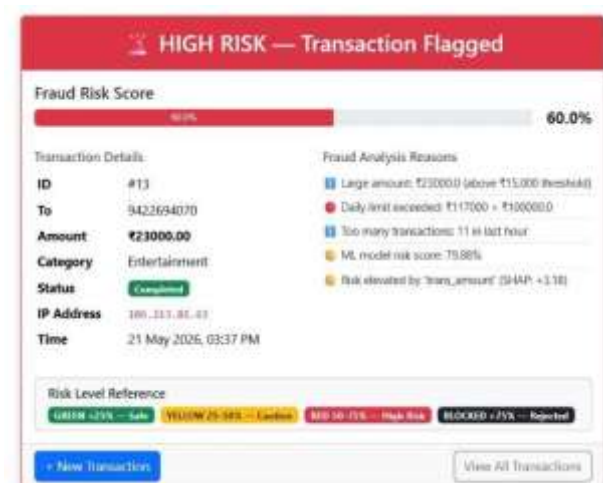
The "SAFE — Transaction Approved" screen shows a "Fraud Risk Score" of 20.0%. The transaction details include ID #4, To: 9957000001, Amount: ₹500.00, Category: Food, Status: Completed, IP Address: 100.113.101.63, and Time: 21 May 2026, 03:08 PM. The "Fraud Analysis Reasons" list includes: "All rule-based checks passed", "ML model risk score: 75.75%", "Receiver UPI 9957000001 linked to IP: 55.179.146.29", "Risk elevated by 'trans_amount' (SHAP: +1.09)", "Risk elevated by 'state' (SHAP: +0.05)", and "Receiver UPI verified SAFE in database". A "Risk Level Reference" bar shows Green (0-25% Safe), Yellow (25-50% Caution), Red (50-75% High Risk), and Blocked (75%+ Rejected). Buttons for "New Transaction" and "View All Transactions" are at the bottom.

- Caution/ Medium Risk Transaction:



The "CAUTION — Medium Risk Detected" screen shows a "Fraud Risk Score" of 30.0%. The transaction details include ID #7, To: 8007500300, Amount: ₹15000.00, Category: Transfer, Status: Completed, IP Address: 100.113.101.63, and Time: 21 May 2026, 03:13 PM. The "Fraud Analysis Reasons" list includes: "Medium-high amount: ₹15000.00 (above ₹8,000 threshold)", "High-risk category: 'transfer' has high fraud rate (2.76%)", "ML model risk score: 77.39%", and "Risk elevated by 'trans_amount' (SHAP: +3.10)". The "Risk Level Reference" bar is the same as in the Safe Transaction screen. Buttons for "New Transaction" and "View All Transactions" are at the bottom.

- High Risk:



The "HIGH RISK — Transaction Flagged" screen shows a "Fraud Risk Score" of 60.0%. The transaction details include ID #13, To: 9422694070, Amount: ₹23000.00, Category: Entertainment, Status: Completed, IP Address: 100.113.101.63, and Time: 21 May 2026, 03:37 PM. The "Fraud Analysis Reasons" list includes: "Large amount: ₹23000.00 (above ₹15,000 threshold)", "Daily limit exceeded: ₹117000 > ₹100000.0", "Too many transactions: 11 in last hour", "ML model risk score: 79.88%", and "Risk elevated by 'trans_amount' (SHAP: +3.18)". The "Risk Level Reference" bar is the same as in the previous screens. Buttons for "New Transaction" and "View All Transactions" are at the bottom.

- Blocked Transaction:



The backend uses SQLAlchemy ORM for database management, WTForms for input validation, and Flask-Migrate for schema handling. The integrated system delivers real-time risk assessment within milliseconds, combining ML intelligence with an intuitive and secure user interface.

6. CONCLUSION

The PaySure project provides a comprehensive solution for real-time fraud detection in UPI-based digital transactions. By employing advanced machine learning techniques, particularly a highly optimized CatBoost gradient boosting classifier, the system can effectively identify anomalous and potentially fraudulent activities with high accuracy. The integration of real-time monitoring ensures that users are promptly alerted to suspicious transactions, thereby reducing financial risk and improving overall trust in digital payment systems. The adoption of the Agile development methodology allowed for iterative improvements, quick feedback incorporation, and progressive enhancement of both the ML models and the system's user interface. This approach ensured that the system could adapt to changing fraud patterns and evolving user requirements. Moreover, the use of CRISP-DM for structured data mining and model development ensured that the solution was data-driven, reliable, and scalable. In summary, PaySure not only strengthens the security infrastructure of UPI payment systems but also serves as a flexible and extensible framework that can be applied to other digital payment platforms. The project highlights the importance of combining AI techniques with best software development practices to achieve effective, real-time fraud detection.

REFERENCES

- [1] M. Anjali, M. Ajay, M. Saiteja, and B. O. Yadav, "Paysafe ai: Intelligent fraud detection for upi transactions using machine learning," International Journal of Innovative Research in Computer and Communication Engineering, pp. 112–118, 2025.
- [2] R. A. Kumar, S. Ishrat, M. D. Prasad, P. A. Siddiq, and N. C. H. Shankar, "Ai-driven detection mechanism for upi fraud and qr code tampering," in IEEE International Conference on Computational Intelligence and Data Science, pp. 330–337, 2025.
- [3] D. M. Rao, T. R. S. Varun, T. V. Gupta, and D. M. Kumar, "Comparative analysis of upi fraud detection using ensemble learning," International Journal of Engineering and Advanced Technology, pp. 452–458, 2025.
- [4] Rethisha.R and S. Cyindia.H, "Leveraging machine learning techniques for real-time detection of upi fraud," International Research Journal of Modernization in Engineering Technology and Science, pp. 502–509, 2025.
- [5] G. H. Trinity and N. Sharma, "Unified payments interface (upi): Fraud and prevention strategies," Journal of Banking and Financial Technology, pp. 88–94, 2025.
- [6] N. Kumar, K. Hemavathi, S. G. K. Sravani, T. Aprna, and M. Deepthi, "Enhancing upi security through deep learning-based fraud detection," International Journal of Research Publication and Reviews, pp. 212–219, 2025.
- [7] S. Bharath, G. L. V. Prasad, V. Sujatha, S. Hemajothi, D. Mani, and N. G. Merlin, "Hmlm: An intelligent artificial intelligence assisted strategy to identify upi frauds based on hybrid markov learning methodology," 2024.
- [8] V. Gupta, S. Sharma, S. Pathak, and S. Nimkar, "Upi based financial fraud detection using deep learning approach," 2024.
- [9] M. Tamilselvi, J. Giri, and D. Sheela, "Experimental evaluation unified payment interface (upi) fraud de tection system using elevated deep learning methodology," 2024.
- [10] R. Rani, A. Alam, and A. Javed, "Secure upi: Machine learning-driven fraud detection system for upi transactions," 2024.
- [11] D. Devassy, D. K. S, G. P. Mannickathan, A. Biju, A. T, and D. R, "Fedupi: Federated learning empowered detection of fraudulent upi transactions," 2025.

- [12] K. S. A, P. M, and S. J, “Enhancing upi fraud detection accuracy using isolation forest: A novel machine learning approach,” in 2025 International Conference on Emerging Technologies in Engineering Applications (ICETEA), 2025.