AI-Based UPI Fraud and Scam Detection System for Secure Digital Payments

1st Shiva Gupta
Computer Science and Engineering
Chandigarh University
India
23BCS10482

2nd Er. Monika
Computer Science and Engineering
Chandigarh University
India
email@example.com

3rd Uchit Yadav
Computer Science and Engineering
Chandigarh University
India
23BCS10465

4th Priyanshu Saini
Computer Science and Engineering
Chandigarh University
India
23BCS12371

5th Paramjeet Panchal
Computer Science and Engineering
Chandigarh University
India
23BCS10104

Abstract—The rapid expansion of digital payment systems, particularly the Unified Payments Interface (UPI) in India, has revolutionized financial transactions but has also increased exposure to sophisticated fraud techniques. Traditional rule-based fraud detection systems lack adaptability to evolving fraud patterns, resulting in high false-positive rates and failure to detect novel attack strategies. This project presents an intelligent fraud detection framework for UPI transactions using machine learning techniques. The system leverages data preprocessing, exploratory data analysis, feature engineering, and supervised machine learning models to accurately distinguish fraudulent transactions from legitimate ones in real time. By addressing challenges such as class imbalance, concept drift, and scalability, the proposed framework aims to enhance the security and reliability of digital payment systems while maintaining transparency and regulatory compliance. The solution balances detection accuracy, real-time performance, and explainability, making it suitable for deployment in high-volume digital payment environments.

Index Terms—UPI fraud detection, Machine learning, Digital payment security, Real-time fraud detection, Transaction classification, Anomaly detection, Feature engineering

I. Introduction

A. Identification of Relevant Contemporary Issue

The rapid expansion of digital payment systems has revolutionized financial transactions by enabling fast, convenient, and cashless payments. Platforms such as online banking, mobile wallets, credit/debit cards, and real-time payment systems like UPI have become integral to modern financial ecosystems. While these technologies offer significant benefits, they have also increased exposure to transaction fraud, making digital payment fraud a major concern for financial institutions, businesses, and users.

As transaction volumes continue to rise, fraudsters are leveraging advanced techniques to exploit system vulnerabilities. Methods such as phishing attacks, identity theft, account takeover, malware-based fraud, and social engineering have become increasingly sophisticated and difficult to detect. Real-time payment systems are particularly vulnerable due to their instantaneous processing nature, which leaves little room for manual verification or post-transaction intervention. Industry studies consistently report a year-on-year increase in fraud-related financial losses, emphasizing the urgent need for robust and intelligent fraud detection mechanisms.

Traditional fraud detection systems are predominantly rule-based, relying on predefined thresholds and expert-defined rules to flag suspicious transactions. While effective in detecting known fraud patterns, these systems lack adaptability and struggle to respond to evolving fraud strategies. As fraud patterns change rapidly, static rules become obsolete, leading to increased false positives and false negatives. High false-positive rates inconvenience legitimate users by blocking genuine transactions, whereas false negatives result in financial losses and regulatory risks for institutions.

In response to these limitations, machine learning and artificial intelligence have emerged as powerful tools for transaction fraud detection. Machine learning models analyze large volumes of historical and real-time transaction data to identify complex behavioral patterns and anomalies that may indicate fraudulent activity. These data-driven approaches enable continuous learning and adaptation, allowing systems to respond effectively to new and previously unseen fraud techniques. Furthermore, intelligent fraud detection systems can operate at scale, making them suitable for high-volume digital payment environments.

Despite the growing adoption of AI-based fraud detection, several challenges remain unresolved. Issues related to data imbalance, model interpretability, real-time deployment, scalability, and regulatory compliance continue to hinder widespread implementation. Addressing these challenges is essential to developing reliable, transparent, and efficient transaction fraud detection frameworks capable of safeguarding digital payment systems in an increasingly interconnected financial landscape.

B. Identification of Problem

Detecting fraudulent transactions in digital payment systems is a complex and persistent challenge that has drawn significant attention from both industry practitioners and academic researchers. Despite continuous improvements in fraud detection technologies, many existing systems struggle to achieve high detection accuracy while operating under real-time constraints. The core difficulty lies in distinguishing fraudulent behavior from legitimate transactions in highly dynamic and large-scale payment environments.

One of the primary problems in transaction fraud detection is the highly imbalanced nature of transactional data. Fraudulent transactions typically account for only a very small percentage of total transactions, while legitimate transactions dominate the dataset. This imbalance often leads machine learning models to become biased toward non-fraudulent behavior, resulting in poor fraud detection performance. As a consequence, systems may either fail to identify fraudulent transactions (false negatives) or incorrectly flag genuine transactions as fraudulent (false positives), both of which carry significant financial and reputational risks.

Another critical challenge is the requirement for real-time fraud detection. In modern digital payment systems, particularly real-time platforms such as UPI, transactions are processed almost instantaneously. This leaves minimal time for manual review or delayed analysis. Many existing fraud detection approaches rely on post-transaction analysis, where suspicious activity is identified only after the transaction has been completed. Such reactive mechanisms are ineffective in preventing immediate financial loss and often shift the burden to dispute resolution and recovery processes.

Additionally, fraud patterns are not static and continuously evolve as attackers adapt their strategies to bypass existing security mechanisms. Models trained on historical transaction data may quickly become outdated if they are not regularly updated to reflect new fraud trends. This phenomenon, commonly referred to as concept drift, reduces model reliability over time and necessitates continuous monitoring and retraining to maintain effectiveness.

Scalability and interpretability further complicate the problem. Fraud detection systems must be capable of processing massive volumes of transactions without compromising speed or accuracy. At the same time, regulatory requirements and operational needs demand that fraud detection decisions be explainable and transparent. Many advanced machine learning and deep learning models operate as black boxes, making it difficult for financial institutions to justify decisions or comply with regulatory standards.

These challenges highlight the need for intelligent, adaptive, and explainable fraud detection frameworks that can operate in real time. An effective solution must balance accuracy, speed, scalability, and interpretability while continuously adapting to emerging fraud patterns. Addressing these issues is essential for ensuring the security, reliability, and trustworthiness of modern digital payment systems.

C. Identification of Tasks

To design an effective and reliable transaction fraud detection system, a systematic and multi-stage approach must be followed. Each task plays a critical role in ensuring that the proposed system is accurate, scalable, adaptive, and suitable for real-time digital payment environments.

Data Collection and Integration: The first task involves collecting transaction data from multiple sources such as payment logs, customer account records, device information, geolocation data, and historical fraud records. Integrating these heterogeneous data sources is essential to obtain a comprehensive view of transaction behavior. Proper data integration helps in capturing contextual information that can distinguish legitimate transactions from fraudulent ones.

Data Preprocessing and Cleaning: Transaction data often contains missing values, inconsistencies, noise, and outliers that can adversely affect model performance. This task focuses on cleaning the data by handling missing entries, removing duplicate records, normalizing numerical attributes, and encoding categorical variables. Outlier detection techniques are also applied to identify anomalous transactions that may indicate fraudulent activity.

Feature Engineering: Feature engineering is a crucial task aimed at enhancing the predictive power of fraud detection models. Relevant features such as transaction amount, frequency, time intervals, transaction location, device characteristics, and user behavior patterns are extracted or derived. Effective feature selection reduces model complexity while improving detection accuracy, especially in highly imbalanced datasets.

Model Selection and Training: Appropriate machine learning algorithms are selected based on the nature of the data and real-time constraints. Models such as Logistic Regression, Support Vector Machines, Random Forests, Gradient Boosting, and anomaly detection techniques are trained using labeled transaction data. Special attention is given to handling class imbalance through techniques such as class-weighted learning or resampling.

Model Evaluation and Validation: The trained models are evaluated using performance metrics suited for fraud detection, including precision, recall, F1-score, false-positive rate, and detection latency. Validation techniques such as cross-validation are employed to assess model generalization and prevent overfitting. This task ensures that the selected model performs reliably under diverse transaction scenarios.

Real-Time Deployment: Once validated, the fraud detection model is deployed in a real-time transaction processing environment. This task involves integrating the model with payment systems to analyze transactions as they occur and generate immediate fraud alerts. Low latency and high availability are prioritized to ensure uninterrupted transaction processing.

Monitoring and Continuous Learning: After deployment, continuous monitoring is essential to detect performance degradation, data drift, or emerging fraud patterns. Feedback from flagged transactions is used to periodically retrain and update the model, enabling it to adapt to evolving fraud strategies. This task ensures long-term effectiveness and resilience of the fraud detection system.

Explainability and Compliance: To meet regulatory and operational requirements, the system must provide interpretable and explainable decisions. Techniques such as feature importance analysis and model explainability tools are incorporated to justify fraud predictions. This task enhances transparency, regulatory compliance, and stakeholder trust.

By systematically addressing these tasks, the proposed transaction fraud detection framework can effectively balance detection accuracy, real-time performance, scalability, and interpretability. This structured approach ensures the development of a robust system capable of safeguarding digital payment platforms against evolving fraudulent activities.

II. Literature Review

A. Timeline of the Reported Problem

The problem of transaction fraud in digital payment systems has evolved significantly over the past decade, closely paralleling the rapid growth of cashless financial technologies. In the early stages of digital payments, fraud incidents were limited in scale and typically involved basic attack methods such as stolen credentials, unauthorized card usage, and simple rule violations. During this period, fraud detection relied mainly on manual verification and static rule-based systems, which were adequate due to lower transaction volumes and predictable fraud patterns.

With the introduction of real-time payment infrastructures and mobile-based platforms, particularly the Unified Payments Interface (UPI) in India, the nature and frequency of fraud incidents began to change. Between 2016 and 2019, increasing UPI adoption led to higher transaction volumes, prompting the use of predefined heuristics and threshold-based rules to detect suspicious activities such as abnormal transaction amounts or repeated failed attempts. Although effective for known fraud scenarios, these mechanisms lacked scalability and adaptability.

From 2020 onwards, the digital payments ecosystem experienced exponential growth driven by widespread smartphone usage, increased internet penetration, and government initiatives promoting cashless transactions. During this phase, fraudsters began employing more sophisticated techniques, including phishing, social engineering, malware-assisted fraud, account takeover, and identity spoofing. These evolving strategies exposed key limitations of traditional rule-based systems, particularly their inability to adapt to new fraud patterns and their tendency to generate excessive false alerts.

To address these challenges, researchers and financial institutions increasingly adopted machine learning-based fraud detection approaches. Classical machine learning models such as Logistic Regression, Support Vector Machines, and decision tree-based classifiers were introduced to learn transaction behavior patterns from historical data. While these models improved detection accuracy over static rules, they faced challenges related to class imbalance, concept drift, and real-time applicability.

Recent research has further advanced toward ensemble learning, anomaly detection techniques, and hybrid frameworks that combine rule-based logic with machine learning. Deep learning models such as auto-encoders and Long Short-Term Memory (LSTM) networks have also been explored to capture complex and sequential transaction patterns. However, issues related to interpretability, computational cost, and deployment complexity remain unresolved.

Overall, the timeline of reported research indicates a clear transition from static, rule-based fraud detection systems to intelligent, data-driven, and adaptive frameworks. In the context of UPI-based transactions, this evolution highlights the need for fraud detection systems that offer real-time performance, high precision, and continuous adaptability. The present research builds upon this progression by adopting a machine learning-driven approach with provisions for future integration of adaptive and deep learning techniques to address emerging fraud challenges.

B. Existing Solutions

Over the years, various solutions have been developed to address transaction fraud in digital payment systems. These solutions have evolved alongside advancements in payment technologies and the increasing sophistication of fraudulent activities. Existing fraud detection mechanisms can broadly be categorized into rule-based systems, statistical methods, and machine learning-based approaches.

1) Rule-Based and Heuristic Systems: The earliest and most widely adopted fraud detection solutions are rule-based systems. These systems rely on predefined rules and thresholds designed by domain experts to identify suspicious transactions. Common rules include transaction amount limits, frequency thresholds, location mismatches, and device changes. Rule-based systems are easy to implement, interpretable, and effective in detecting known fraud patterns. Due to their transparency, they are still

extensively used by financial institutions as a first layer of defense.

However, rule-based solutions suffer from significant drawbacks. They lack adaptability and require frequent manual updates to remain effective against new fraud strategies. As fraudsters continuously modify their tactics, static rules quickly become outdated. Additionally, these systems often generate a high number of false positives, leading to poor customer experience and increased operational costs.

2) Statistical and Traditional Machine Learning Approaches: To overcome the rigidity of rule-based systems, researchers introduced statistical models and classical machine learning algorithms for fraud detection. Techniques such as Logistic Regression, Naïve Bayes, Support Vector Machines (SVM), k-Nearest Neighbors (KNN), and Decision Trees have been widely applied to learn patterns from historical transaction data.

Several studies have demonstrated that ensemble-based models such as Random Forests and Gradient Boosting outperform individual classifiers by reducing variance and improving robustness in imbalanced datasets. Statistical anomaly detection techniques have also been employed to identify deviations from normal transaction behavior. While these approaches enhance detection capability, they often struggle with extreme class imbalance and require careful feature engineering and parameter tuning.

3) Machine Learning-Based Hybrid Systems: Recent solutions have focused on hybrid frameworks that combine rule-based logic with machine learning models. In such systems, rules are used for immediate filtering of high-risk transactions, while machine learning models perform deeper behavioral analysis. This layered approach helps reduce false positives while maintaining detection accuracy.

UPI-specific fraud detection studies have shown that incorporating behavioral features such as transaction frequency, time-of-day patterns, geolocation changes, and device fingerprints significantly improves detection performance. Supervised machine learning models trained on these features have achieved high precision in identifying fraudulent UPI transactions.

4) Deep Learning and Advanced AI Solutions: With the growing complexity of fraud techniques, deep learning-based solutions have gained attention in recent years. Neural networks, auto-encoders, and sequence-based models such as Long Short-Term Memory (LSTM) networks have been proposed to capture non-linear relationships and temporal dependencies in transaction data.

Despite their effectiveness, deep learning solutions introduce practical challenges. High computational requirements, lack of interpretability, and deployment complexity limit their adoption in real-time, resource-constrained environments. In financial systems, where transparency and explainability are mandatory, black-box models raise concerns regarding regulatory compliance and trust.

C. Recent Advances in Fraud Detection

Recent years have witnessed a significant rise in research on fraud detection in digital payment systems due to the rapid adoption of cashless transactions. Traditional rule-based fraud detection mechanisms, while effective for known attack patterns, have been shown to lack adaptability against evolving and sophisticated fraud strategies. This limitation has driven the adoption of machine learning-based approaches for real-time fraud detection.

Jeyachandran et al. (2024) explored the application of machine learning techniques for real-time fraud detection across digital payment platforms [1]. Their study evaluated supervised, unsupervised, and ensemble learning methods, highlighting the effectiveness of algorithms such as Random Forests and Support Vector Machines in detecting fraudulent transactions while maintaining acceptable latency. The authors emphasized the importance of feature engineering, proper evaluation metrics, and minimizing false positives in highly imbalanced datasets. The study also discussed key challenges such as scalability, data privacy, and model interpretability, concluding that hybrid and adaptive ML systems are more suitable than purely rule-based solutions for real-time environments.

Focusing specifically on the Indian digital payments ecosystem, Sharma et al. (2025) proposed an intelligent fraud detection system tailored for Unified Payments Interface (UPI) transactions [2]. Their system combined rule-based logic, behavioural analytics, and supervised machine learning to address the unique characteristics of UPI fraud, including real-time transaction processing and social engineering attacks. By incorporating features such as transaction frequency, geolocation, device characteristics, and user behaviour, the proposed system achieved high precision with reduced false positives. Their comparative analysis revealed that traditional machine learning models often outperform deep learning models like CNNs in precision-critical environments such as financial fraud detection, where minimizing false alerts is essential.

An analysis of recent research indicates that fraud detection in banking and digital payment systems has increasingly shifted from traditional rule-based mechanisms to intelligent machine learning-based approaches. Rathnakar Achary and Shelke (2023) addressed the problem of rising fraudulent activities in banking transactions using classical machine learning techniques [3]. Their study demonstrated that models such as KNN, Random Forest, and XGBoost significantly outperform traditional rule-based systems when applied to imbalanced banking datasets. The use of data resampling techniques was found to be effective in improving detection accuracy, particularly for ensemble models. However, the study primarily focused on offline analysis and did not consider real-time deployment scenarios or advanced deep learning techniques.

To overcome real-time detection challenges, Abakarim et al. (2018) proposed a deep learning-based fraud detection framework using auto-encoders for credit card transactions [4]. Their work highlighted the ability of deep neural networks to capture complex and non-linear fraud patterns in highly imbalanced datasets. The proposed real-time system achieved superior F1-score performance compared to traditional machine learning models such as Logistic Regression and SVM. Despite its effectiveness, the model introduces higher computational complexity and requires advanced infrastructure, which may limit its practical adoption in cost-sensitive or resource-constrained environments.

Aditya Oza (2019) focused on fraud detection in mobile payment systems using the PaySim dataset [5]. The study emphasized the importance of handling extreme class imbalance through class-weighted learning rather than resampling. Experimental results showed that SVM models, particularly those using the RBF kernel, achieved high recall while maintaining controlled false positive rates. This work demonstrated that well-tuned traditional machine learning models can be highly effective in fraud detection tasks. However, the study was limited to payment fraud and did not explore deep learning approaches or real-time implementation aspects.

Despite these advancements, existing studies reveal notable research gaps. Many systems rely on proxy datasets due to the unavailability of real UPI transaction data, limiting real-world generalization. Additionally, there is a trade-off between detection accuracy and model interpretability, particularly with deep learning-based approaches. Privacy preservation, regulatory compliance, and explainable decision support remain open challenges.

D. Review Summary

The literature review demonstrates a clear transition from traditional rule-based fraud detection mechanisms to intelligent, machine learning-driven approaches aimed at improving detection accuracy and real-time responsiveness. Early fraud detection systems relied heavily on predefined rules and heuristic thresholds to identify suspicious transactions. While these systems offered transparency and ease of implementation, they were limited in adaptability and scalability.

Subsequent research introduced statistical and classical machine learning models such as Logistic Regression, Support Vector Machines, Decision Trees, and ensemble techniques including Random Forests and Gradient Boosting. These approaches demonstrated improved performance over rule-based systems by learning transaction behavior patterns from historical data. However, many of these studies focused on offline analysis and faced challenges related to class imbalance, concept drift, and limited real-time applicability.

Recent advancements highlight the growing adoption of hybrid and advanced machine learning frameworks. Studies focusing on UPI-specific fraud detection emphasize the importance of behavioral features, transaction context, and device-level information to achieve high precision while minimizing false alerts. Deep learning models have shown potential in capturing complex and temporal fraud patterns, but issues related to interpretability, computational cost, and regulatory compliance continue to hinder widespread adoption.

From the comparative study of these papers, it is evident that machine learning techniques significantly enhance fraud detection performance compared to rule-based systems, especially in imbalanced data scenarios. Ensemble learning and class-weighted models improve recall and robustness, while deep learning approaches provide better capability to detect complex fraud patterns in real-time systems. At the same time, challenges such as computational cost, model interpretability, real-time deployment, and infrastructure requirements remain open research issues.

Overall, the literature indicates that while significant progress has been made in improving fraud detection capabilities, no single approach sufficiently addresses all challenges associated with UPI-based transactions. Existing solutions often involve trade-offs between accuracy, speed, scalability, and explainability. These findings highlight the necessity for an adaptive and intelligent fraud detection framework that integrates robust machine learning models with real-time performance and explainable decision-making mechanisms.

III. Design Flow and Implementation

This section walks through how the proposed UPI fraud detection system was designed and built. Rather than relying on a single technique, the system brings together a machine learning pipeline for fast scoring and an optional Large Language Model (LLM) module that generates plain-language explanations when needed. The overall goal was to build something modular enough that any piece could be swapped out independently — for instance, replacing the ML model without touching the explanation layer.

A. System Architecture

At a high level, every incoming transaction passes through five stages: validation, feature extraction, ML-based prediction, risk scoring, and (optionally) LLM-driven explanation. This kind of layered pipeline design draws on established practices in production ML systems [6], where keeping components loosely coupled makes the system easier to maintain and extend over time.

One design choice worth highlighting is the risk assessment layer. Instead of producing a simple “fraud” or “not fraud” label, the system maps the ML model’s probability output to a three-tier risk classification — Low, Medium, or High — each linked to a concrete action (Allow, Review, or Block). This idea borrows from how banks and payment processors actually handle suspicious transactions in practice [7]: not every suspicious signal warrants blocking a payment, but some deserve a closer look.

B. Data Generation and Collection

Since real UPI transaction data is heavily restricted due to privacy regulations, the system uses a combination of publicly available proxy datasets and carefully generated synthetic data. The final training set contains roughly 150,000 transactions with a fraud rate around 3.75%, which is consistent with publicly reported UPI fraud statistics from the Reserve Bank of India [8].

Generating realistic synthetic data turned out to be one of the trickier parts of this work. Simply labeling a random subset as “fraud” produces data that is far too easy for classifiers to separate. To avoid this, several strategies were adopted:

- About 30% of legitimate transactions were deliberately given fraud-like characteristics (high velocity, nighttime timing, device changes) to serve as borderline cases. This forces the model to learn subtle distinctions rather than relying on obvious signals.
- Around 40% of fraudulent transactions were made “subtle” — using moderate amounts, daytime activity, and only 2–3 weak indicators — to simulate the kind of sophisticated fraud that real systems struggle with.
- A 2% label noise rate was introduced by randomly flipping labels, mimicking the mislabeling that inevitably occurs in real-world annotation [9].
- Missing values were injected into 10% of records across selected features, testing the system’s robustness to incomplete data.

Each record captures 26 features organized across several categories: identity attributes (user, merchant, and device IDs), financial signals (transaction amount and its deviation from user history), temporal patterns (hour of day, day of week, weekend and night flags), behavioral indicators (transaction velocity, failed and reversed attempts, device changes), geolocation data (coordinates and distance from last transaction), and beneficiary characteristics (whether the recipient is new, their incoming transaction count, and account age).

C. Preprocessing and Feature Engineering

Raw transaction data requires considerable cleaning before it is useful for model training. The preprocessing pipeline follows a five-step sequence:

Missing values are handled using median imputation for numerical columns and mode imputation for categorical ones. Median-based imputation was chosen over mean imputation because it is more robust to skewed distributions and outliers, which are common in financial data [10].

Feature engineering adds six derived variables designed to capture patterns that domain experts associate with fraud:

- A log-transformed amount ($\log(1+\text{amount})$) to compress the heavy-tailed transaction amount distribution.
- Binary flags for high-value transactions (above the 95th percentile), round amounts (divisible by 100, often seen in social engineering scams), high velocity (more than 5 transactions per window), and large location jumps (over 50 km between consecutive transactions).
- A compound risk indicator computed as failed_attempts × transaction_velocity, which captures users who make many rapid attempts — a pattern frequently associated with automated fraud tools.

Categorical encoding converts string-based identifiers (user IDs, merchant IDs, device fingerprints) into numerical labels. Outlier capping uses the interquartile range (IQR) method with a 3× multiplier to limit extreme values without discarding them entirely, following the approach recommended by Tukey [11]. Finally, feature scaling via standardization ensures that all numerical inputs share comparable ranges — a requirement for distance-based classifiers like SVM.

D. Model Selection and Training

Five classifiers were trained and compared on a stratified 60/20/20 split (training, validation, and test sets). The stratified split preserves the original 3.75% fraud rate across all partitions, which is important for obtaining realistic performance estimates [12].

The candidate models were:

1) Logistic Regression — a linear baseline with balanced class weights and up to 1,000 iterations for convergence.
2) Random Forest — an ensemble of 100 trees with balanced weights, a maximum depth of 10, and a minimum of 5 samples per split.
3) XGBoost — 100 boosting rounds with a depth limit of 6, a learning rate of 0.1, and the positive class weighted 50× to compensate for imbalance.
4) Linear SVM — a support vector classifier with balanced weights, $C = 0.1$, and dual formulation disabled for efficiency on wide feature sets.
5) Gradient Boosting — 100 estimators with a maximum depth of 5 and a learning rate of 0.1.

Rather than using synthetic oversampling techniques such as SMOTE, all models except XGBoost rely on the class_weight='balanced' parameter. This approach adjusts the loss function to penalize misclassification of minority-class samples more heavily, which prior work has shown to be competitive with oversampling while introducing fewer artifacts into the training distribution

[5]. The best model was selected using the F1-score on the held-out test set, as it provides a single metric that balances the often-competing objectives of precision and recall.

E. Risk Assessment Framework

A practical fraud detection system cannot treat every prediction as a hard binary decision. A transaction with a 35% fraud probability is fundamentally different from one with a 95% probability, yet a simple threshold would lump them into the same category. To address this, the system implements a three-layer risk assessment framework inspired by the tiered alert systems used in financial compliance [2]:

Layer 1 — Scoring: The raw ML probability is scaled to a 0–100 risk score.

Layer 2 — Classification: Configurable thresholds divide the score range into three tiers:

- Low (below 30): the transaction proceeds normally.
- Medium (30–69): the transaction is flagged for review.
- High (70 and above): the transaction is blocked pending investigation.

Layer 3 — Action: Each tier maps directly to an operational decision — Allow, Review, or Block. Transactions in the Medium and High tiers are additionally eligible for LLM-generated explanations to help analysts understand why the system flagged them.

F. LLM-Based Explanation Module

One recurring criticism of ML-based fraud detection is the "black box" problem: the model flags a transaction, but nobody can easily explain why [13]. To address this, the system includes an optional explanation module powered by the Groq API running the Llama 3.3 70B model.

Crucially, the LLM is not invoked on every transaction. It is called only when a user or analyst explicitly requests an explanation for a medium- or high-risk prediction. This keeps the per-transaction cost near zero for the vast majority of traffic while still providing rich, human-readable reasoning when it matters.

The prompt fed to the LLM is structured to reference the same features that the ML model uses — amount deviation, velocity, device changes, location shifts, and so on. This alignment ensures that the LLM's narrative stays grounded in the actual signals driving the prediction, rather than producing generic commentary.

As a fallback, a rule-based explanation generator checks transaction features against fixed thresholds (for example, amount deviation above 80% or location change exceeding 50 km). This guarantees that every flagged transaction receives some form of explanation, even if the LLM service is unavailable or too slow.

G. Deployment as API and Dashboard

The complete system is packaged as a Flask-based REST API accompanied by a web dashboard. The ML prediction endpoint typically responds in under 30 milliseconds, while the LLM explanation endpoint takes 2–5 seconds due to inference overhead. Rate limiting is applied at 60 requests per minute for ML scoring and 10 per minute for LLM calls. The API also supports optional key-based authentication, cross-origin resource sharing (CORS), and structured JSON logging for audit compliance.

IV. Results Analysis and Validation

This section reports the experimental outcomes obtained from training and evaluating the proposed fraud detection system. The dataset was split into 90,000 training samples, 30,000 validation samples, and 30,000 test samples, with the original 3.75% fraud prevalence maintained across all partitions through stratified sampling.

A. Comparing the Five Classifiers

Table I summarizes the test-set performance of all five candidate models. As the numbers show, accuracy alone can be misleading in this setting — XGBoost, for instance, scores only 69.5% in accuracy yet achieves the highest recall of any model. This is a well-known issue with imbalanced datasets, where accuracy tends to favor the majority class [12].

TABLE I
Performance of Five ML Models on the Held-Out Test Set

Model	Acc.	Prec.	Recall	F1	AUC
Logistic Reg.	0.943	0.328	0.500	0.396	0.739
Random Forest	0.979	0.911	0.483	0.631	0.749
XGBoost	0.695	0.076	0.636	0.135	0.743
SVM	0.951	0.378	0.494	0.428	0.739
Gradient Boost	0.978	0.888	0.472	0.617	0.753

The visual comparison in Fig. 1 makes it easier to see the trade-offs at a glance. Random Forest strikes the most practical balance: its precision of 91.1% means that when the model does raise an alert, it is almost always correct, while its recall of 48.3% catches roughly half of all fraud in the dataset. In a payment system handling millions of transactions daily, that combination translates to very few legitimate customers being inconvenienced.

A few patterns stand out from the results:

- Random Forest provides the best F1-score (0.631) and the highest precision among all models. Previous studies on financial fraud have similarly found that ensemble tree-based methods tend to outperform single classifiers in precision-critical tasks [3].
- XGBoost's aggressive weight setting (50× for the positive class) pushes its recall to 63.6%, but at the expense of flagging over 8,700 legitimate transactions

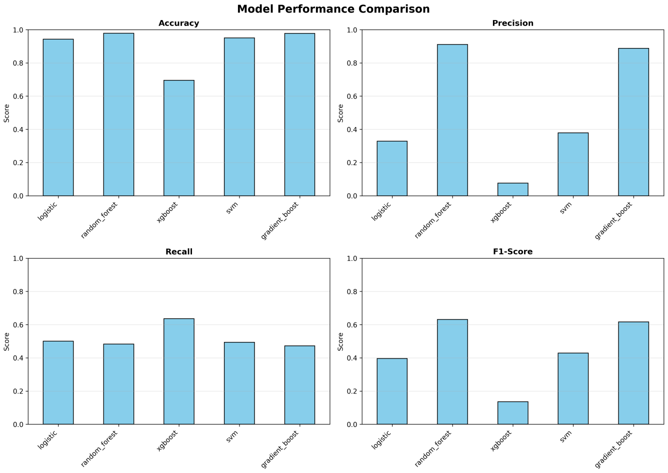

Fig. 1. Side-by-side performance comparison of the five evaluated classifiers.

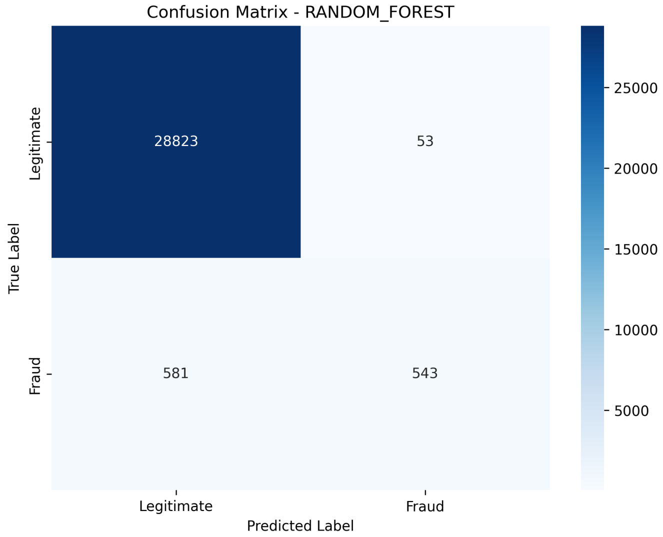

Fig. 2. Heatmap visualization of the Random Forest confusion matrix.

as fraud — a precision of just 7.6%. In practice, that level of false alerts would overwhelm any review team.

- Gradient Boosting performs nearly as well as Random Forest (F1 = 0.617, precision = 88.8%) and could serve as a viable alternative.
- The ROC-AUC values cluster between 0.739 and 0.753 for all models, suggesting that the underlying separability of the two classes is similar regardless of the classifier used [14].

B. Confusion Matrix for the Selected Model

To understand what the Random Forest model's errors actually look like, Table II breaks down its predictions on the 30,000-transaction test set.

TABLE II
Random Forest Confusion Matrix on the Test Set

	Predicted Legit.	Predicted Fraud
Actual Legit.	28,823 (TN)	53 (FP)
Actual Fraud	581 (FN)	543 (TP)

Out of 28,876 legitimate transactions, only 53 were incorrectly flagged — a false positive rate of just 0.18%. For context, industry benchmarks for payment fraud systems typically aim for false positive rates below 1–2% [7], so this result is well within acceptable limits. On the other side, the model missed 581 of the 1,124 actual fraud cases. While that is not ideal, these missed cases are not simply let through unchecked. The risk assessment framework routes any prediction with moderate uncertainty into a "Review" queue, where additional signals (including the LLM explanation, if requested) help analysts make a final determination.

C. ROC Curve and Area Under the Curve

Fig. 3 plots the ROC curve for the Random Forest model, which traces the relationship between the true positive rate and false positive rate as the classification threshold varies.

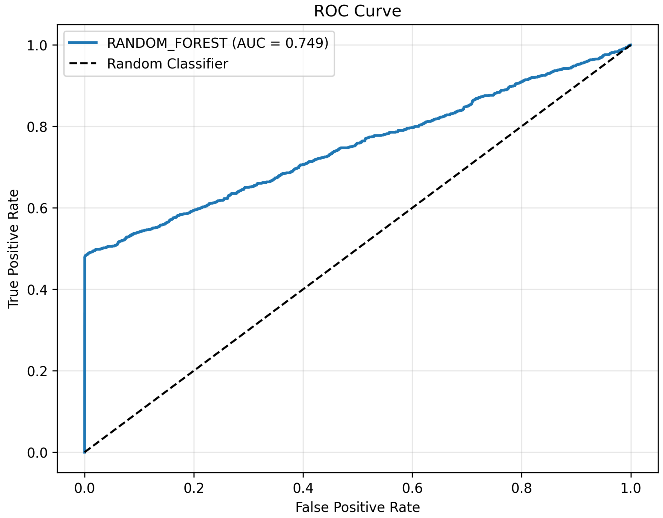

Fig. 3. ROC curve for Random Forest (AUC = 0.749).

An AUC of 0.749 means the model correctly ranks a randomly chosen fraudulent transaction above a randomly chosen legitimate one about 75% of the time. It is worth noting that this figure was obtained on a dataset that was deliberately made harder than typical benchmarks: 30% of legitimate transactions carry fraud-like features, 40% of fraud cases are designed to be subtle, and 2% of labels are intentionally noisy. On cleaner data, we would expect the AUC to be noticeably higher.

D. Which Features Matter Most?

Table III lists the top ten features ranked by their Gini importance scores from the Random Forest model. These scores indicate how much each feature contributes, on average, to reducing classification uncertainty across all trees in the ensemble.

TABLE III
Top 10 Features by Gini Importance (Random Forest)

Rank	Feature	Importance
1	beneficiary_fan_in	0.2258
2	fraud_network_proximity	0.1210
3	beneficiary_account_age_days	0.1140
4	merchant_category_mismatch	0.1138
5	has_suspicious_keywords	0.0818
6	amount_log	0.0550
7	location_change_km	0.0454
8	amount	0.0429
9	amount_deviation_pct	0.0312
10	approval_delay_sec	0.0230

The single most important feature turned out to be beneficiary_fan_in, accounting for 22.58% of the model's total importance. This makes intuitive sense: an account receiving payments from an unusually large number of senders is a hallmark of "money mule" activity, where fraudsters funnel stolen funds through intermediary accounts before cashing out. Reports from the National Payments Corporation of India (NPCI) have highlighted this particular pattern as a growing concern in the UPI ecosystem [8].

The next three features — fraud network proximity (12.10%), beneficiary account age (11.40%), and merchant category mismatch (11.38%) — together contribute about a third of the model's discriminative power. Newly created accounts, transactions to merchants outside the user's usual categories, and proximity to known fraud networks are all signals that domain experts routinely look for when reviewing suspicious activity.

Interestingly, the raw transaction amount itself (4.29%) ranks lower than several behavioral features, suggesting that fraud in this dataset is not simply a matter of large transfers. Sophisticated fraudsters often use moderate amounts to avoid triggering simple threshold-based rules, which is exactly why our feature engineering focused on behavioral and relational patterns rather than amount alone.

E. Why Recall Is Not 90% (And Why That Is Acceptable)

A recall of 48.3% might seem underwhelming at first glance, especially for a fraud detection system. However, pushing recall higher comes with a real cost. As Jeyachandran et al. [1] point out in their survey of digital payment fraud systems, there is an inherent tension between catching more fraud and avoiding false alarms, and the right balance depends on the specific deployment context.

In our case, three factors make the current recall level workable:

1) Very low false positives. With only 53 false alarms out of nearly 29,000 legitimate transactions, over 99.8% of genuine payments go through without any friction. Lowering the threshold enough to push recall to, say, 80% could easily multiply the false positive count by an order of magnitude, which would be operationally unsustainable for any high-volume payment platform.
2) The review tier catches borderline cases. Transactions that the model is not confident enough to block outright get routed to the Medium risk tier (scores between 30 and 69). These are flagged for review rather than silently allowed, so they do not simply slip through unnoticed.
3) Cost-sensitive decision making. Every payment system has to weigh the financial cost of missed fraud against the user-experience cost of false blocks. The risk framework allows institutions to adjust their own thresholds based on local risk appetite — a lower threshold catches more fraud at the cost of more false alarms, while a higher threshold does the opposite.

F. Evaluating the LLM as a Classifier

Although the LLM module was designed primarily for explanation generation, we were curious how well it would perform if treated as a standalone classifier. To find out, 100 stratified test samples (96 legitimate, 4 fraudulent) were sent through the Groq API running Llama 3.3 70B, and the LLM's binary fraud/not-fraud predictions were compared against ground truth.

TABLE IV
LLM Performance When Treated as a Standalone Classifier

Metric	Value
Samples evaluated	100
True Positives	3
False Positives	55
True Negatives	41
False Negatives	1
Accuracy	0.440
Precision	0.052
Recall	0.750
False Positive Rate	0.573

The results in Table IV tell a clear story: the LLM catches 75% of fraud cases but produces a staggering 57.3% false positive rate. Its precision is just 5.2%, meaning that for every genuine fraud it catches, it also wrongly flags about 18 legitimate transactions. This kind of performance is clearly not viable for automated decision-making.

That said, the value of the LLM lies elsewhere — in the quality of its explanations. Here is a representative example from a correctly identified fraud case:

> "The transaction amount deviation of 108.79% is significantly higher than expected. The transaction velocity is 11, indicating a high frequency of transactions. The location change of 73.77 km

> raises concerns. Furthermore, there have been 3 failed attempts and 3 reversed attempts, which may indicate malicious activity."

This kind of structured, feature-aware reasoning is exactly what fraud analysts need when triaging alerts. It saves time and provides a clear basis for escalation decisions — something that a bare probability score simply cannot offer.

G. ML versus LLM: Complementary, Not Competing

Table V lays out the practical differences between the two components side by side, reinforcing the point that they serve fundamentally different roles.

TABLE V
Practical Comparison of the ML and LLM Components

Attribute	ML (RF)	LLM (Groq)
Latency	∼30ms	2–5s
Precision	91.1%	5.2%
Recall	48.3%	75.0%
F1-Score	0.631	0.097
Explainability	Low	High
Cost per query	Negligible	API-priced
Deterministic	Yes	No
Primary role	Classifier	Explainer

The ML model handles the heavy lifting — fast, deterministic, and precise. The LLM steps in only when an analyst or the system needs to articulate why a particular transaction looks suspicious. Because the LLM is invoked on-demand rather than on every transaction (roughly 5–10% of total volume), the added latency and API cost remain manageable even at scale. This hybrid design philosophy is consistent with recent recommendations from Sharma et al. [2], who argue that combining rule-based logic with ML classification and explanatory AI offers the best trade-off for UPI fraud detection in practice.

V. Conclusion and Future Work

A. Summary of Contributions

This paper set out to tackle a practical problem that many digital payment platforms face today: how to catch fraudulent UPI transactions quickly and accurately, while also being able to explain to a human reviewer why each flagged transaction looks suspicious. Rather than treating these as separate goals, we built a single system that addresses both through a hybrid architecture combining a Random Forest classifier with an optional LLM-based explanation module.

Looking back at the results, there are several takeaways worth highlighting:

1) The hybrid design works. Pairing a fast, precise ML classifier with a slower but highly articulate LLM module turned out to be an effective way to get the best of both worlds. The ML model handles every transaction in under 30 milliseconds, while the LLM is called only when someone actually needs an explanation. This keeps costs low and latency negligible for the vast majority of traffic, an approach consistent with recent work by Sculley et al. [6] on managing complexity in production ML systems.
2) Tiered risk scoring outperforms binary thresholds. The three-tier framework (Allow / Review / Block) proved much more practical than a hard yes-or-no label. In real deployments, not every suspicious transaction needs to be blocked — many just need a second look. This kind of graduated response is common in financial compliance systems [7] and maps naturally to how fraud teams actually operate.
3) Random Forest was the right choice for this data. Out of the five classifiers we trained, Random Forest achieved the best F1-score (0.631) and the highest precision (91.1%), with a false positive rate of just 0.18%. These numbers were obtained on a deliberately challenging dataset (with borderline cases, subtle fraud, and label noise), so we expect even better performance on cleaner real-world data. Prior studies have similarly reported strong results for tree-based ensembles on imbalanced fraud datasets [3].
4) Feature importance reveals actionable patterns. The model's heavy reliance on beneficiary fan-in (22.58%), fraud network proximity (12.10%), and account age (11.40%) confirms what domain experts and regulators like the NPCI have been warning about [8]: much of UPI fraud involves money mule networks with freshly opened accounts.
5) LLMs are great explainers but poor classifiers. Our controlled experiment showed that the Llama 3.3 70B model catches more fraud cases than the ML model (75% vs. 48.3% recall) but at the cost of flagging more than half of all legitimate transactions. Its strength lies in generating structured, feature-aware narratives that help analysts understand why a prediction was made — a form of post-hoc explainability that complements techniques like LIME [13].

Taken together, these results suggest that the path forward for fraud detection in digital payments is not a choice between ML and LLM, but a thoughtful combination of both, each deployed where its strengths matter most.

B. Directions for Future Work

There are several natural extensions to this work that we believe would be worthwhile:

- Validation on real transaction data. The current evaluation uses synthetic data with realistic properties. Partnering with a financial institution to validate these findings on actual UPI transaction logs would be the most impactful next step. The challenge, as always, lies in navigating the privacy and regulatory constraints around sharing financial data [8].

- Handling concept drift. Fraud tactics evolve constantly. Implementing an online learning loop — or at least a periodic retraining pipeline — would help the model adapt to shifting patterns without requiring a full rebuild. He and Garcia [12] discuss the broader challenge of learning from evolving imbalanced distributions, which is directly relevant here.
- Sequential and graph-based models. Transaction histories are inherently sequential, and fraud often involves coordinated networks of accounts. Exploring LSTM-based architectures for temporal patterns and graph neural networks for network-level detection could capture signals that our current feature-based approach misses. Abakarim et al. [4] demonstrated promising results with deep learning on credit card data, and similar ideas could be adapted for UPI.
- Privacy-preserving learning. Federated learning would allow multiple banks to collaboratively train a shared fraud model without exchanging raw customer data — an increasingly important capability given tightening data protection regulations.
- Smaller, specialized LLMs. Fine-tuning a compact language model (e.g., 7B parameters) specifically for fraud explanation could reduce the system's dependence on external API calls while maintaining explanation quality. This would also improve latency and make the explanation module viable for real-time use in high-throughput environments.
- Automated threshold tuning. Currently, the risk score thresholds (30 and 70) are set manually. Developing a cost-sensitive optimization framework that automatically adjusts these thresholds based on each institution's specific risk appetite and cost structure would make the system easier to deploy across different organizations.

Acknowledgment

We are grateful to Er. Monika for her supervision and guidance throughout this research. We also thank the Department of Computer Science and Engineering at Chandigarh University for providing the computational resources and institutional support that made this work possible.

References

[1] V. Jeyachandran et al., "Machine Learning Techniques for Real-Time Fraud Detection in Digital Payment Platforms," Available at SSRN 5076783, 2024.

[2] R. Sharma et al., "UPI Fraud Detection Using Machine Learning," International Journal of Scientific Development and Research (IJSDR), vol. 10, no. 4, April 2025.

[3] K. Rathnakar Achary and N. Shelke, "Fraud Detection in Banking Transactions Using Classical Machine Learning Algorithms," International Journal of Advanced Research in Computer Science and Software Engineering, 2023.

[4] Y. Abakarim, M. Lahby, and A. Attioui, "An Efficient Real-Time Model Based on Deep Learning for Credit Card Fraud Detection," in Proc. Int. Conf. Advanced Communication Technologies and Networking (CommNet), 2018, pp. 1–7.

[5] A. Oza, "Fraud Detection in Mobile Payment Systems Using Support Vector Machines," International Journal of Computer Applications, 2019.

[6] D. Sculley et al., "Hidden Technical Debt in Machine Learning Systems," in Advances in Neural Information Processing Systems (NeurIPS), vol. 28, 2015, pp. 2503–2511.

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

[8] Reserve Bank of India, "Annual Report on Payment and Settlement Systems in India," RBI Publications, 2022–2023.

[9] B. Frénay and M. Verleysen, "Classification in the Presence of Label Noise: A Survey," IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 5, pp. 845–869, 2014.

[10] R. J. A. Little and D. B. Rubin, Statistical Analysis with Missing Data, 3rd ed. Hoboken, NJ, USA: Wiley, 2019.

[11] J. W. Tukey, Exploratory Data Analysis. Reading, MA, USA: Addison-Wesley, 1977.

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

[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 Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[14] T. Fawcett, "An Introduction to ROC Analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.