IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, March 2026
Impact Factor: 8.2

An Integrated UPI Payment System with Pre-Transaction AI Fraud Detection, NFC Tap-to-Pay, and Offline QR Payment Support

Shivam Devendra More[1] and Reena Gharat[2]
Student, Department of Computer Technology[1]
Guide, Department of Computer Technology[2]
Bharati Vidyapeeth Institute of Technology, Kharghar, Maharashtra, India

Abstract: *Digital payment adoption in India has grown at a pace that existing fraud prevention mechanisms struggle to match. Most consumer-facing UPI applications detect suspicious activity after a transaction is already processed, leaving users exposed at the most critical moment. This paper presents a UPI-based Android payment application that runs a machine learning fraud evaluation directly on the device before the user confirms any payment, so potential risks are identified before money moves. The system bundles a trained Random Forest classifier — exported to ONNX format and loaded as an Android asset — that classifies each pending payment as normal, suspicious, or fraudulent using nine behavioral transaction features, with no network request required at inference time. In addition to fraud prevention, the application supports NFC tap-to-pay through Android's Host Card Emulation service and provides a QR-based offline payment mode for devices that do not support NFC, ensuring broad hardware compatibility. A dedicated B2B module enables merchant order management and business transactions within the same platform. An administrative React dashboard provides centralized oversight of flagged users, reported transactions, and system activity. The two-tier architecture — Android application with bundled ONNX model and a Supabase backend — delivers a complete, layered payment ecosystem that addresses security, accessibility, and operational usability within a single system..*

Keywords: UPI, Fraud Detection, Random Forest, ONNX, On-Device Inference, NFC, Offline Payment, QR Code, Android, B2B Transactions, Supabase

I. INTRODUCTION

India's digital payment landscape has changed considerably since the Unified Payments Interface was introduced by the National Payments Corporation of India. What began as a mechanism for simple bank-to-bank transfers has expanded into an infrastructure that supports consumer payments, merchant transactions, government disbursements, and business-to-business settlements across billions of transactions every month [1]. The speed and convenience of UPI have driven adoption across income groups and geographies, making it one of the most widely used real-time payment systems in the world.

Alongside this growth, payment fraud has become an increasingly serious problem. Attackers have adapted their methods to exploit the trust that users place in digital payment applications — using phishing links, social engineering, account takeover, and fake merchant setups to redirect funds. A common characteristic of these attacks is that they succeed precisely because the victim acts quickly, without any signal from the application that something is wrong. Most UPI platforms only review transactions for fraud after they have been processed, which means a user who has been manipulated into sending money typically has no recourse.

The application described in this paper takes a different approach. Before the user reaches the MPIN confirmation screen, a trained machine learning model evaluates nine features describing the pending transaction entirely on the

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, March 2026 Impact Factor: 8.2

device. The model — a Random Forest classifier converted to ONNX format and bundled directly as an Android asset — classifies the transaction as normal, suspicious, or fraudulent without any network call. The result is shown to the user before confirmation is possible, making fraud detection a visible, pre-transaction step rather than a background post-processing activity.

Beyond fraud prevention, the application addresses two practical limitations that affect a significant portion of the target user base. Near-field communication support, which enables tap-to-pay payment, is absent on many Android handsets in the mid-range and budget segments that dominate the Indian market [3]. The application handles this by providing a fully functional offline QR payment mode as an alternative, so users without NFC-capable devices are not excluded from contactless payment. The application also includes a B2B merchant module that allows registered merchants to manage orders, generate invoices, and handle business transactions through the same interface — a capability that no existing lightweight UPI application currently offers in an integrated form.

The system is built across two layers: an Android application written in Java — which bundles the ONNX fraud model for on-device inference — and a Supabase backend that handles database storage, authentication, and real-time data synchronization. A React administrative dashboard connects to the same backend to provide centralized system management.

II. LITERATURE REVIEW

The trajectory of UPI adoption in India has been studied from multiple angles. Kolte and Humbe [1] examined the uptake of UPI and BHIM across different user segments and found that the simplicity of virtual payment address-based transactions and the speed of fund settlement were the key reasons users switched from traditional payment methods. Their analysis also draws attention to the widening gap between how quickly UPI transactions are processed and how slowly fraud response systems catch up — a structural weakness that the pre-transaction, on-device detection approach in this paper directly targets.

Machine learning for financial fraud detection has been an active area of research for over a decade. Xuan et al. [2] conducted a comparative evaluation of classification algorithms on credit card transaction data and demonstrated that Random Forest outperformed logistic regression, support vector machines, and individual decision trees on skewed datasets where fraudulent samples are a small minority. Their work specifically noted the importance of class weighting and ensemble depth in achieving reliable performance on imbalanced data — findings that shaped the configuration of the fraud detection model used in this project, which uses balanced class weights and 300 estimators to handle the natural rarity of fraudulent transactions.

Near-field communication technology is increasingly used as the foundation for contactless payment implementations. Hassan [3] conducted a comprehensive review of security vulnerabilities in NFC systems, covering eavesdropping, relay attacks, and data modification risks, alongside software and protocol-level countermeasures. This review informed the decision to implement NFC payment in this project using Android's Host Card Emulation service rather than hardware secure elements, which reduces device dependency while maintaining security appropriate for the transaction values typical of UPI payments.

Offline payment capability has received growing research attention as a means of extending financial access to users in areas with poor or intermittent connectivity. Sonawane, Pawar, and Solanki [4] developed BitSync, a mobile application that enables secure peer-to-peer payments using QR codes with no internet connection required. Their approach relies on a two-phase QR protocol with local cryptographic verification and deferred server synchronization, which is directly relevant to the offline payment module in this project — where locally signed QR payloads are stored in a Room database until connectivity is available for sync with the Supabase backend.

On the broader question of machine learning in financial fraud prevention, Omogbeme et al. [5] reviewed the role of big data analytics and ML-based systems across fintech platforms and found that behavioral features — transaction frequency, temporal patterns, and receiver reputation — consistently outperformed simple amount-based thresholds in fraud classification. Their findings align with the feature design in this project, where features such as

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, March 2026 Impact Factor: 8.2

txn_count_last_24h, receiver_is_flagged, and contact_status_encoded contribute meaningfully to model decisions alongside the transaction amount.

III. EXISTING SYSTEM

Several digital payment platforms currently serve Indian consumers and merchants, each with a defined feature set and specific constraints that the proposed system aims to address.

Google Pay and PhonePe are the two most widely used UPI applications in India. Both offer smooth bank account linking, QR-based merchant payments, and contact-based money transfers. Their transaction flows are well optimized, and their user bases are large. However, neither platform provides any real-time fraud evaluation at the moment of transaction initiation. Fraud detection, where it exists, runs at the backend after the transaction has been processed. Users receive no in-app warning before sending money to a suspicious or flagged recipient.

Paytm offers a broader product including a digital wallet, merchant payment acceptance, and limited financial services. Its fraud monitoring is similarly post-transaction, and the platform does not provide an integrated B2B order or invoice management module — merchant interactions are treated as payment receipts rather than structured business transactions.

Enterprise payment platforms such as Zeta and Setu offer more sophisticated fraud monitoring pipelines, but these are built for financial institutions operating at scale, requiring significant infrastructure investment and backend integration.

The limitations common across these systems include: no consumer-facing UPI application provides pre-transaction fraud evaluation visible to the user at the point of payment; NFC-based contactless payment is offered without any offline fallback for devices that lack NFC hardware; B2B transaction management is either absent or disconnected from the core payment interface; and administrative oversight tools require direct backend access rather than a usable dashboard for non-technical operators.

IV. PROPOSED METHODOLOGY

The proposed system is organized into two primary tiers — an Android mobile application and a Supabase backend — with a React administrative dashboard providing management access to the backend layer. Fraud detection runs entirely on the Android device using a bundled ONNX model, requiring no network call at prediction time.

System Architecture and User Modules

The Android application is the primary interface for both regular users and merchants. User registration follows a sequential flow: mobile number entry, OTP verification, personal detail submission, UPI ID selection, bank account linking, and MPIN configuration. Once authenticated, users access a navigation structure that covers home, contacts, transaction history, QR scanner, and profile sections.

Merchants are routed to a separate home screen after login, which surfaces the B2B order management interface and a merchant-specific transaction history. The two user types share the same authentication pipeline and Supabase backend but operate through different UI layers. Local session data is stored using a Room database, and the SessionManager utility handles authentication state across activity transitions.

On-Device Pre-Transaction Fraud Detection

The fraud detection model is a Random Forest classifier trained using scikit-learn in Python. The training dataset contains 100,000 synthetic transaction records constructed to reflect realistic fraud distribution — 75 percent normal transactions, 17 percent suspicious, and 8 percent fraudulent. An 18 percent noise rate was applied during data generation so that the model learns decision boundaries from imperfect, real-world-like data rather than clean synthetic patterns.

Nine features are computed from each transaction before it is evaluated by the model. The raw transaction amount and its ratio against the user's historical average capture absolute and relative size signals. Hour of day and a derived binary

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, March 2026
Impact Factor: 8.2

night flag capture temporal patterns, since high-value transactions occurring between 10 PM and 5 AM are strongly associated with fraud in the training data. On the receiver side, the model evaluates whether the receiver is a known contact, how many fraud reports have been filed against them, and whether they carry an administrative flag. Transaction count in the past 24 hours captures payment velocity, which is a recognized signal for account takeover scenarios. A three-level contact status encoding distinguishes trusted, normal, and reported contacts.

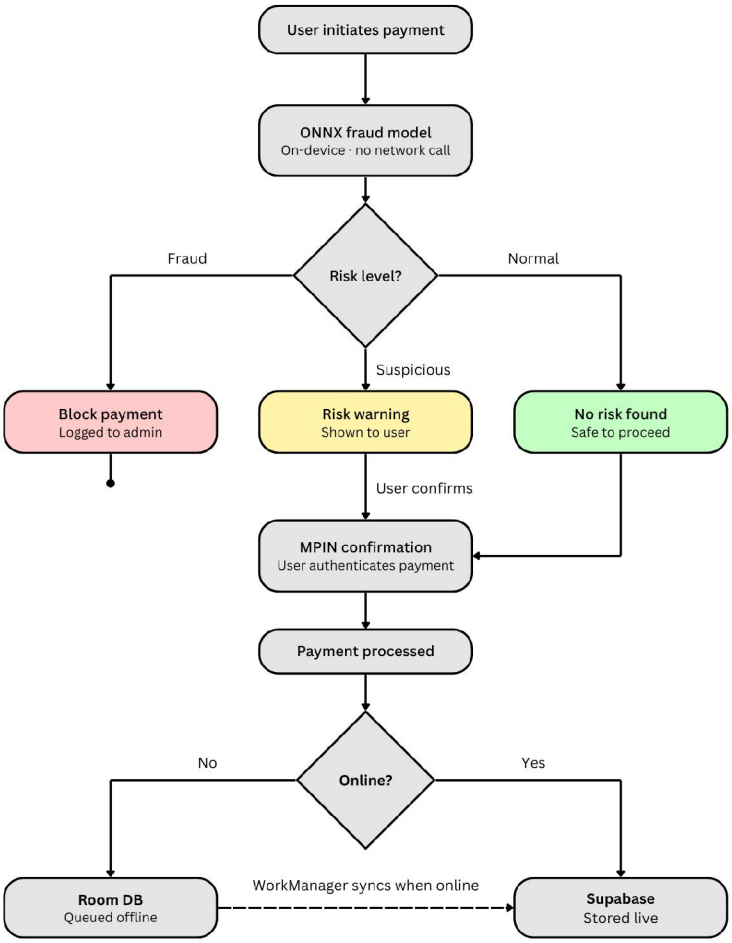

Fig. 1. System architecture overview showing on-device ONNX inference within the Android layer

The model is configured with 300 decision trees, a maximum depth of 12, balanced class weights to compensate for the natural rarity of fraud labels, and out-of-bag scoring for validation during training. After training in Python, the model is converted from the serialized .pkl format to ONNX using skl2onnx, producing a self-contained fraud_model.onnx file. This ONNX file is bundled directly inside the Android application under the assets directory and loaded at runtime using the Microsoft ONNX Runtime for Android library. Because inference runs entirely on the device, no network request is made at prediction time — the fraud check has zero dependency on internet connectivity and adds no network latency to the payment confirmation flow.

When a user initiates a payment, the FraudEngine utility collects the nine feature values from local transaction context, runs the ONNX model on-device through the ONNX Runtime session, and receives a three-class probability array. A normal result allows the payment to proceed to MPIN confirmation. A suspicious result presents a warning screen with

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, March 2026
Impact Factor: 8.2

details of the risk factors before the user can continue. A fraud result blocks the transaction and logs it in Supabase for administrative review.

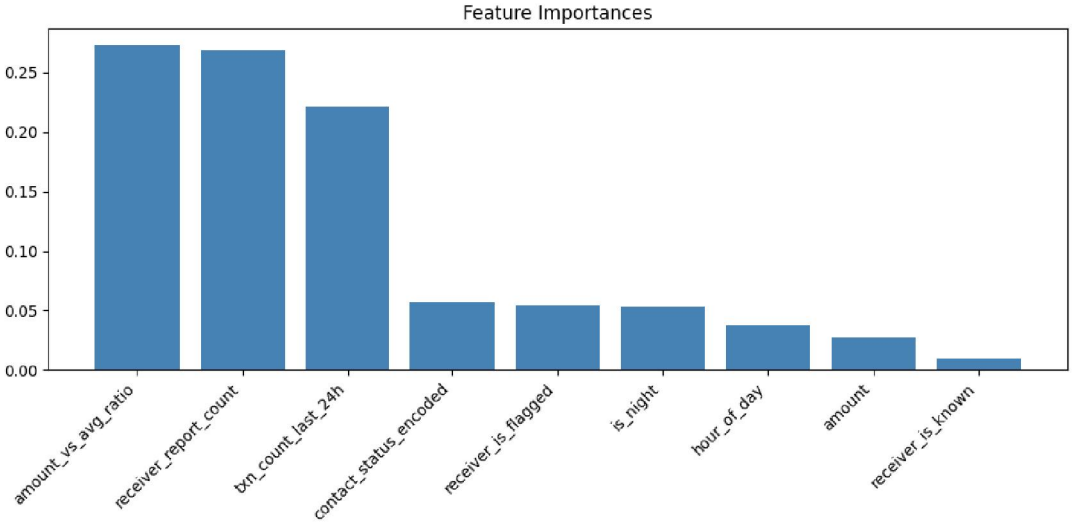

Fig. 2. Feature importances showing relative contribution of each behavioral attribute to fraud classification

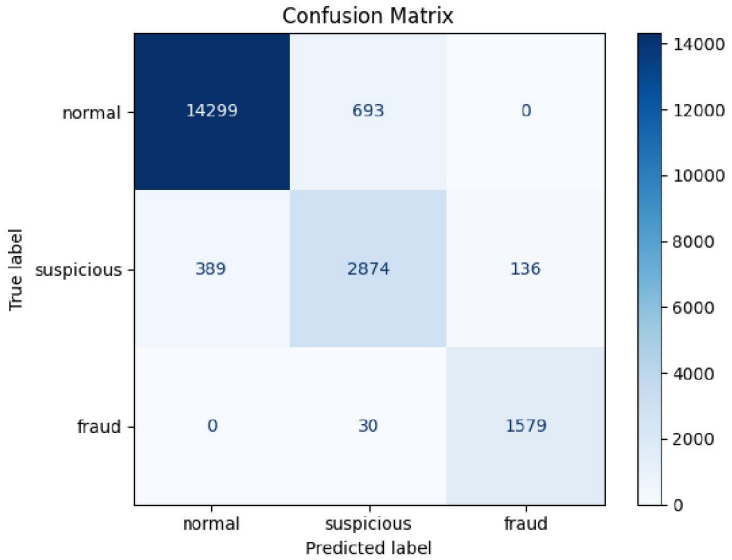

Fig. 3. Confusion matrix evaluated on 20,000 hold-out test samples (20% of total dataset)

NFC Tap-to-Pay and Offline QR Fallback

NFC payment is implemented through Android's Host Card Emulation service. When a user initiates an NFC payment, the application loads the transaction payload into the HCE service, which broadcasts it through the device's NFC controller. The receiving device reads the payload and processes the payment through the standard verification flow. HCE was chosen over hardware secure element approaches because it removes the dependency on SIM-embedded or device-embedded secure chips, making it compatible with a wider range of Android handsets [3].

For users on devices without NFC hardware — a limitation affecting a substantial portion of budget and mid-range Android handsets common in the Indian market — the application provides a complete offline QR payment mode [4].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, March 2026 Impact Factor: 8.2

The sender constructs a signed QR payload containing the receiver's UPI ID, amount, and a timestamp hash. The receiver scans the QR to acknowledge the transaction. The entire exchange requires no active internet connection. Transactions completed in offline mode are stored locally in the Room database and synchronized with the Supabase backend by a WorkManager background task that executes automatically once connectivity is restored.

B2B Merchant Module

The B2B module gives merchants a structured transaction layer beyond simple payment receipt. Registered merchants can browse other merchants on the platform, initiate business orders with line-item detail, generate invoices from completed orders, and track order status through a tabbed interface. Order and invoice data are stored in Supabase and reflected in both the Android application and the administrative dashboard in real time. Merchant home screens display pending orders, recent activity, and a summary of incoming payments, giving business users operational visibility that general-purpose UPI applications do not provide.

Administrative Dashboard

The React administrative panel connects to the Supabase backend and provides structured views of system activity for non-technical administrators. The flagged users page lists accounts that have crossed fraud score thresholds based on on-device model outputs logged to Supabase and manual reports. The reported transactions page shows payments that users have flagged as suspicious, with transaction detail and reporter information. The reported contacts page aggregates contact-level fraud reports to surface receivers who appear across multiple complaints, enabling proactive flagging. The user search function allows lookup by UPI ID or registered mobile number. A bill simulator module supports testing of billing flows in isolation.

V. CONCLUSION

This paper described a UPI-based Android payment application that brings pre-transaction on-device fraud evaluation, NFC contactless payment, offline QR fallback, and B2B merchant management together in a single integrated system. The central contribution — running a trained Random Forest classifier bundled as an ONNX asset directly on the Android device before the user confirms payment — represents a meaningful shift from how consumer-facing UPI applications currently handle fraud prevention. By eliminating the need for any network call during inference, the fraud check operates regardless of connectivity, adds no latency, and gives users a risk assessment when it is still actionable.

The offline QR mode extends payment access to Android device users who are excluded from NFC-based contactless payment due to hardware constraints — a practical concern in the Indian market where budget-segment devices make up a large share of the installed base. The B2B module addresses a genuine gap: merchants using existing UPI applications have no native tools for order tracking or invoice generation within the payment interface, and the proposed system provides both.

Future directions include replacing the synthetic training dataset with real anonymized transaction records to improve model generalization, adding location-based signals as an additional fraud feature, integrating a live payment gateway for actual fund settlement, and extending the application to iOS for broader platform coverage.

REFERENCES

[1]. D. M. Kolte and V. R. Humbe, "Study of UPI/BHIM Payment System in India," International Journal of Science and Research (IJSR), vol. 9, no. 11, pp. 1604–1610, 2020.

[2]. S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random Forest for Credit Card Fraud Detection," in Proc. 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China, Mar. 2018, pp. 1–6, doi: 10.1109/ICNSC.2018.8361343.

[3]. R. Hassan, "Near Field Communication (NFC) Technology: Security Vulnerabilities and Countermeasures," International Journal of Engineering & Technology, vol. 7, no. 4.31, pp. 298–305, 2018.

[4]. P. Sonawane, P. Pawar, and S. Solanki, "BitSync: Enabling Secure Offline Payments with QR Codes," Journal of Advance and Future Research (JAAFR), vol. 3, no. 11, pp. 235–245, Nov. 2025.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, March 2026
Impact Factor: 8.2

[5]. A Omogbeme, I. Atoyebi, A. Soyele, and E. Ogunwobi, "Enhancing Fraud Detection and Prevention in Fintech: Big Data and Machine Learning Approaches," World Journal of Advanced Research and Reviews (WJARR), vol. 24, no. 2, pp. 2301–2319, 2024, doi: 10.30574/wjarr.2024.24.2.3617