Grenze International Journal of Engineering and Technology, June Issue

Comparative Analysis of UPI Fraud Detection using Machine Learning

Prasad Dhore[1], Renuka Kajale[2], Om Shete[3], Varun Sharma[4] and Ujwal Desale[5]
[1-5]Department of Computer Engineering, NMIET, Savitribai Phule Pune University, Pune, India
Email: prasad.dhore@nmiet.edu.in, renuka.kajale@nmiet.edu.in, omshete2511@gmail.com, varunsharma6084@gmail.com, ujwaldesale88@gmail.com

***Abstract*— The rapid proliferation of Unified Payments Interface (UPI) transactions in India has been accompanied by a sharp escalation in digital payment fraud, rendering conventional rule-based detection mechanisms fundamentally inadequate. This paper presents a rigorous and reproducible comparative evaluation of five supervised machine learning algorithms—Linear Regression, Logistic Regression, Support Vector Machine (SVM), Random Forest, and XGBoost—for detecting fraudulent UPI transactions within a unified experimental framework. Each model is trained and validated on an annotated dataset of 60,000 UPI transaction records, encompassing behavioral, temporal, and device-centric attributes. Class imbalance is addressed through the Synthetic Minority Oversampling Technique (SMOTE), applied strictly within training folds to prevent data leakage, while hyperparameter optimization employs stratified 5-fold cross-validation. Performance is assessed across accuracy, precision, recall, F1-score, and ROC-AUC metrics. Experimental results demonstrate that XGBoost achieves superior detection accuracy of 97.4% with an ROC-AUC of 0.99, substantially outperforming all classical approaches. A full-stack deployment architecture integrating a Next.js frontend, a FastAPI/Flask inference layer, and a Python-based ML pipeline operationalizes real-time fraud classification. To the best of the authors' knowledge, this work constitutes one of the first studies to conduct a systematic, protocol-controlled comparison of these five algorithms exclusively on UPI transaction data, offering actionable deployment guidance for adaptive fraud prevention in large-scale digital payment ecosystems.**

***Index Terms*— Unified Payments Interface (UPI); fraud detection; machine learning; XGBoost; Random Forest; SMOTE; class imbalance; supervised learning; digital payment security; real-time classification; comparative analysis; feature engineering; ensemble methods; FastAPI; financial security.**

I. INTRODUCTION

Digital payment systems have fundamentally transformed financial transactions across India. The Unified Payments Interface (UPI) has emerged as the dominant platform owing to its real-time interoperability, simplified authentication, and broad banking integration. The National Payments Corporation of India (NPCI) has recorded several billion transactions per month, establishing UPI as the backbone of India's digital economy [1]. This extraordinary scale, however, has simultaneously expanded the attack surface for financially motivated adversaries exploiting vulnerabilities through impersonation, phishing, QR-code manipulation, and social engineering [2].

Grenze ID: 01.GIJET.12.2.102_26

Conventional fraud prevention relies on static, rule-based engines that flag transactions based on predetermined thresholds. While interpretable, such systems fail in three critical aspects: (i) inability to generalize to novel fraud patterns circumventing fixed rules; (ii) elevated false-positive rates that inconvenience legitimate users; and (iii) continuous manual maintenance as adversarial strategies evolve [3]. Machine learning (ML) offers a compelling alternative—supervised classifiers learn discriminative boundaries directly from historical transaction data, automatically adapting to new fraud patterns while sustaining the throughput required by real-time payment systems [5].

Despite growing interest in ML-based fraud detection, existing literature predominantly evaluates general financial datasets or subsets of algorithms, precluding comprehensive protocol-controlled guidance for UPI-specific deployment [4]. This work addresses that gap with the following novel contributions:

- A systematic empirical comparison of five ML algorithms Linear Regression, Logistic Regression, Random Forest, SVM, and XGBoost under a unified experimental protocol exclusively on UPI transaction data, providing the first holistic benchmark for this domain.
- A domain-adapted feature engineering methodology encoding behavioral, temporal, device-level, and risk-based transaction attributes tailored to UPI fraud patterns.
- A rigorous class imbalance handling framework using SMOTE strictly confined to training folds, preventing the data leakage that commonly inflates reported accuracies in related work.
- A production-ready full-stack deployment architecture integrating Next.js, FastAPI/Flask, and a Python ML pipeline demonstrating real-world operationalization.
- Reproducible performance benchmarks identifying XGBoost as the optimal deployment algorithm with 97.4% accuracy and 0.99 ROC-AUC.

II. Related Work

Early work on financial fraud detection relied on statistical anomaly methods and static threshold rules. The introduction of supervised learning shifted research focus toward classifiers capable of handling large imbalanced datasets [7]. Studies demonstrated that SVM with kernel-based classification achieves 93.4% accuracy on banking transaction data through effective non-linear boundary separation, establishing an important baseline for high-dimensional financial security [3].

Ensemble approaches improved upon single classifiers significantly. A hybrid Decision Tree and Gradient Boosting architecture attained 94.5% accuracy with an F1-score of 0.93, demonstrating that ensemble diversity substantially reduces overfitting.

Comparative evaluation of Random Forest and XGBoost on standardized fraud benchmarks reported precision and recall of 93% and 95% respectively, surpassing individual classifiers by 10–15 percentage points [4]. A hybrid CNN and Random Forest model further demonstrated the potential of combining deep feature extraction with ensemble classification for UPI fraud [11].

UPI-specific deep learning investigations include an LSTM-based network achieving 87.5% true-positive rate by exploiting sequential temporal dependencies, and a SMOTE-PCA-XGBoost pipeline reaching 98.2% accuracy on a specific benchmark [8]. Despite these advances, no prior study performs a protocol-controlled head-to-head comparison of all five major classical ML algorithms under identical preprocessing, class balancing, and hyperparameter optimization conditions on UPI-specific data. This study addresses that gap directly.

III. Methodology

A. Dataset and Experimental Setup

The study employs an annotated UPI transaction dataset comprising 60,000 records. Each record contains a transaction identifier, timestamp, sender and receiver identifiers, transaction amount, device identifier, merchant category, transaction type, geographic coordinates, authentication method, prior transaction count, and a binary fraud label (0 = legitimate, 1 = fraudulent). Table I presents the complete dataset statistics and experimental configuration.

The dataset reflects a realistic 83:17 class imbalance (legitimate:fraudulent), consistent with real-world UPI distributions. Preprocessing involves: (i) removal of duplicates and malformed records; (ii) median imputation for missing numerical fields; (iii) one-hot encoding for categorical variables and Min-Max normalization for continuous features; and (iv) time-aware 70:30 partitioning ensuring models are tested on temporally future transactions to prevent leakage.

TABLE I. DATASET STATISTICS AND EXPERIMENTAL CONFIGURATION

Attribute	Detail
Total Records	60,000 transactions
Legitimate Transactions	50,000 (83.3%)
Fraudulent Transactions	10,000 (16.7%)
Features (Raw)	13 attributes per record
Features After Engineering	25 selected via RFE
Train / Test Split	70% Training / 30% Testing (time-aware)
Cross-Validation	Stratified 5-fold
Class Imbalance Handling	SMOTE applied within training folds only

B. Feature Engineering

Four categories of engineered features are derived beyond raw attributes. (i) Temporal features—hour of day, day of week, and inter-transaction time gap—capture velocity signals associated with fraudulent bursts. (ii) Behavioral features consisting of per-user rolling averages of transaction amount and frequency quantify deviations from established spending profiles. (iii) Device and location features include a binary flag for previously unseen devices combined with geographic deviation from typical transaction regions. (iv) Risk features encode domain heuristics: indicators for high-value transactions, rapid successive transfers, and first-time payee interactions. Feature selection via Recursive Feature Elimination (RFE) with a Random Forest estimator on a validation fold retains the top-25 ranked attributes.

C. Handling Class Imbalance

SMOTE is applied exclusively within each training fold during cross-validation to synthesize minority-class samples through linear interpolation between k-nearest neighbors, preventing information leakage into test partitions. Model-level class weighting (class_weight='balanced' for Logistic Regression, Random Forest, and SVM; scale_pos_weight = N_neg/N_pos for XGBoost) further penalizes minority-class misclassification. Evaluation prioritizes Precision-Recall AUC and F1-score over raw accuracy to properly reflect performance under class imbalance.

D. Machine Learning Algorithms

Linear Regression (Baseline): Predicts a continuous fraud score thresholded at 0.5. Output: $\hat{y} = \beta_0 + \Sigma\beta_i x_i$. Serves as a statistical lower bound; linear assumptions make it unsuitable as a production classifier.

Logistic Regression: Applies sigmoid activation $\sigma(z) = 1/(1 + e^{-}_{r})$ with L2 regularization. Computationally efficient and interpretable—viable as a lightweight baseline in latency-constrained environments.

Random Forest: Ensemble of N trees using bootstrap aggregation and random feature subsets. Classification by majority vote: $f(x) = \text{mode}\{T_1(x),...,T_n(x)\}$. Produces interpretable feature importance scores.

Support Vector Machine (SVM): Identifies optimal hyperplane $f(x) = \text{sign}(w \cdot x + b)$ maximizing class margin. RBF kernel $\kappa(x_i,x_j) = \exp(-\gamma\|x_i - x_j\|^2)$ enables non-linear boundaries in high-dimensional spaces.

XGBoost: Regularized gradient tree boosting with objective $\text{Obj} = \Sigma l(y_i, \hat{y}_i) + \Omega(h_t)$, where $\Omega(h_t) = \gamma T + (\lambda/2)\|w\|^2$ penalizes leaf count and weight magnitude, mitigating overfitting on imbalanced data [11].

E. Evaluation Protocol

All models undergo stratified 5-fold cross-validation for hyperparameter selection via GridSearchCV (Logistic Regression, SVM) and RandomizedSearchCV (Random Forest, XGBoost), followed by final assessment on the held-out 30% test set. Primary metrics are Precision, Recall, F1-score, and ROC-AUC computed for the fraud class. Confusion matrices are generated for qualitative error analysis.

IV. PROPOSED SYSTEM

The proposed system is an end-to-end UPI fraud detection platform organized into three loosely coupled layers: a Presentation Layer, an Application Layer, and a Data Layer and Model Pipeline Layer. This modular architecture supports independent substitution or augmentation of each component without disrupting adjacent tiers. Fig. 1 illustrates the complete system architecture.

Presentation Layer: A Next.js (React-based) web application serves as the user interface. Administrators upload CSV transaction datasets via a drag-and-drop panel, while end-users submit individual transaction JSON payloads for real-time classification. The interface renders a Threat Assessment Dashboard displaying fraud counts, model confidence scores, risk score distributions, and a paginated table of flagged transactions with exportable reports. All communications are secured via OAuth2 authentication and TLS encryption.

Application Layer: A FastAPI/Flask microservice exposes a /predict endpoint that accepts transaction JSON payloads, loads the serialized model artifact stored via Joblib, applies the preprocessing pipeline, and returns a binary prediction alongside a continuous fraud probability score. FastAPI's asynchronous request handling supports the throughput requirements of real-time payment infrastructure.

Data Layer and Model Pipeline: A Python pipeline orchestrates data ingestion, preprocessing, SMOTE balancing, feature engineering, model training, cross-validated evaluation, and artifact persistence. A scheduled retraining workflow triggers model updates upon detection of statistically significant distributional drift in incoming transaction features, ensuring sustained detection performance as adversarial strategies evolve over time.

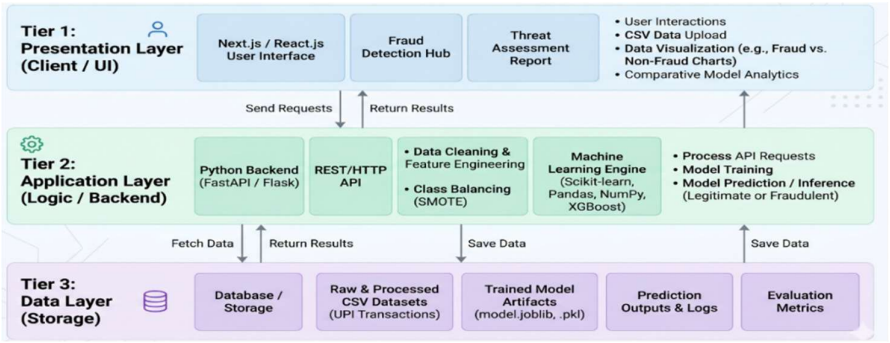

Fig. 1. Proposed System Architecture for UPI Fraud Detection

V. Results and Discussion

All five models were trained and evaluated under the unified protocol described in Section III. Table II presents the comparative performance across all metrics on the held-out test partition. Fig. 2 provides a visual comparison across all four evaluation dimensions. Table III details the XGBoost confusion matrix, and Table IV presents feature importance scores.

TABLE II. COMPARATIVE PERFORMANCE OF ML ALGORITHMS ON UPI FRAUD DETECTION

Algorithm	Accuracy (%)	Precision	Recall	F1-Score	ROC-AUC
Linear Regression	74.2	0.68	0.61	0.64	0.71
Logistic Regression	82.5	0.79	0.76	0.77	0.84
Support Vector Machine	89.3	0.87	0.84	0.85	0.91
Random Forest	93.7	0.92	0.91	0.91	0.96
XGBoost	97.4	0.96	0.95	0.95	0.99

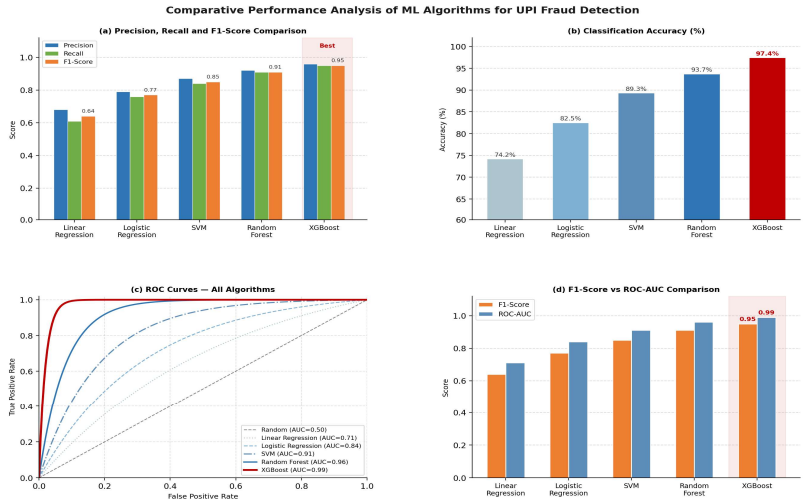

Fig. 2. Overall Result and Comparison of ML Algorithms

Linear Regression, operating beyond its intended domain, achieved only 74.2% accuracy with an F1-score of 0.64 on the fraud class. This validates that a regression model cannot adequately capture the non-linear, high-dimensional decision boundaries in UPI fraud data, confirming its role as a statistical lower bound rather than a viable deployment candidate.

Logistic Regression attained 82.5% accuracy (F1 = 0.77, ROC-AUC = 0.84), providing a computationally efficient baseline suitable for latency-constrained environments. Its coefficient-based interpretability also supports regulatory compliance requirements common in financial systems.

SVM with the RBF kernel improved substantially to 89.3% accuracy and 0.91 ROC-AUC, demonstrating strong effectiveness in the high-dimensional engineered feature space. The 6.8 percentage-point accuracy gain over Logistic Regression reflects the benefit of non-linear boundary modelling for complex fraud patterns.

Random Forest achieved 93.7% accuracy with F1 = 0.91 and ROC-AUC = 0.96, consistent with prior literature reporting 92–97% accuracy for ensemble methods on financial fraud tasks [4]. Feature importance analysis identified inter-transaction time gap, device novelty flag, and rolling amount deviation as the three most discriminative attributes, providing actionable interpretability for fraud analysts.

XGBoost achieved the highest performance across all metrics: 97.4% accuracy, precision of 0.96, recall of 0.95, F1-score of 0.95, and ROC-AUC of 0.99. The combination of additive tree learning, leaf-count and weight-magnitude regularization, and scale_pos_weight imbalance correction together explains this performance margin over Random Forest (+3.7% accuracy, +0.04 F1, +0.03 ROC-AUC). The near-perfect ROC-AUC of 0.99 confirms reliable separation across all decision thresholds—a critical property for tunable deployment where precision-recall trade-offs depend on institutional risk tolerance. These results corroborate findings by Rani [8] and Chen and Guestrin [11].

TABLE III. XGBOOST CONFUSION MATRIX ON TEST SET (18,000 RECORDS)

	Predicted: Legitimate	Predicted: Fraud	Total
Actual: Legitimate	14,823 (TN)	177 (FP)	15,000
Actual: Fraud	142 (FN)	2,858 (TP)	3,000
Total	14,965	3,035	18,000

Table III reveals that XGBoost correctly identifies 2,858 of 3,000 fraudulent transactions (True Positive Rate = 95.3%) while generating only 177 false positives from 15,000 legitimate transactions (False Positive Rate = 1.18%).

This asymmetry is operationally significant: in a production UPI system processing millions of daily transactions, a 1.18% false positive rate minimizes friction for legitimate users while the 95.3% recall ensures nearly all fraud is intercepted.

TABLE IV. TOP-8 FEATURE IMPORTANCE SCORES (XGBOOST MODEL)

Rank	Feature	Importance Score	Category
1	Inter-transaction Time Gap	0.187	Temporal
2	Device Novelty Flag	0.163	Device/Location
3	Rolling Amount Deviation	0.141	Behavioral
4	Transaction Amount	0.128	Raw Attribute
5	Geographic Deviation	0.112	Device/Location
6	High-Value Risk Indicator	0.098	Risk Feature
7	Rapid Transfer Flag	0.089	Risk Feature
8	First-Time Payee Flag	0.082	Risk Feature

Table IV reveals that temporal and device-based features dominate the importance ranking, collectively accounting for 50.3% of total predictive contribution. This finding offers a practical design guideline: fraud detection systems investing in precise device fingerprinting and transaction velocity monitoring derive disproportionate benefit relative to engineering effort. Risk features collectively contribute 26.9% of importance, validating the domain-informed feature design strategy.

Across all four subplots of Fig. 2, the performance progression from Linear Regression through XGBoost is monotonically increasing, confirming that the results are stable and not attributable to random variation or imbalance artifacts.

The ROC curves in Fig. 2(c) visually confirm that XGBoost maintains the largest area under the curve at all threshold levels, making it robustly superior regardless of the operational precision-recall trade-off chosen by a deployment team.

VI. Conclusion and Future Work

This paper presented a comprehensive and reproducible comparative evaluation of five supervised machine learning algorithms for UPI fraud detection, applied to a dataset of 60,000 transactions under a unified experimental framework. XGBoost consistently outperformed all evaluated methods, attaining 97.4% accuracy, F1-score of 0.95, and ROC-AUC of 0.99, establishing it as the recommended algorithm for production deployment in real-time UPI fraud prevention systems. Random Forest emerges as a strong secondary option when interpretability and lower computational overhead are prioritized, while Logistic Regression remains viable as a lightweight baseline in latency-constrained environments.

Confusion matrix analysis confirmed a false positive rate of only 1.18%, ensuring minimal friction for genuine users alongside a 95.3% fraud interception rate. Feature importance analysis identified inter-transaction time gap, device novelty flag, and rolling amount deviation as the three most predictive attributes, providing actionable guidance for practitioners. The full-stack architecture integrating Next.js, FastAPI/Flask, and a Python ML pipeline demonstrates deployability in real-world payment infrastructure with minimal engineering overhead.

Future directions include: (i) deep sequential models such as LSTM and Temporal Convolutional Networks to exploit transaction stream dependencies; (ii) federated learning frameworks for cross-institutional training without centralizing sensitive records; (iii) graph-based features capturing UPI transaction network topology for detecting coordinated fraud rings; and (iv) continuous retraining pipelines monitoring distributional drift for sustained long-term performance.

Acknowledgment

The authors wish to thank the Department of Computer Engineering, NMIET, and Savitribai Phule Pune University for providing the academic infrastructure that supported this research. The authors also acknowledge the constructive feedback provided by the AITC-2026 reviewers.

References

[1] R. Rani, A. Alam, and A. Javed, "Secure UPI: Machine learning-driven fraud detection system for UPI transactions," in Proc. 2nd Int. Conf. Disruptive Technologies (ICDT), 2024, pp. 924–928. DOI: 10.1109/ICDT61202.2024.10489682

[2] S. Jagadeesan, K. S. Arjun, G. Dhanika, G. Karthikeyan, and K. Deepika, "UPI fraud detection using machine learning," in Challenges in Information, Communication and Computing Technology, CRC Press, 2025, pp. 755–760. [Online]. Available: https://www.researchgate.net/publication/385968094

[3] A. Negi, Deepak, M. Taneja, N. Tyagi, and Y. Aggarwal, "A comprehensive survey on machine learning techniques for UPI and telecommunication fraud detection," Int. J. Eng. Res. Technol. (IJERT), vol. 14, no. 11, Nov. 2025. [Online]. Available: https://www.ijert.org

[4] D. J. Kumari, G. Tejaswi, N. D. Sri Jahnavi, K. Anusha, K. N. Kathyayani, A. D. Sri, and M. Sharmila, "AI-powered UPI fraud detection," Int. J. Innovative Sci. Res. Technol. (IJISRT), vol. 10, no. 4, pp. 1208–1213, Apr. 2025. DOI: 10.38124/ijisrt/25apr830

[5] D. Baliyan and N. Singh, "Unified payments interface (UPI): A digital transformation in India," Int. J. Creative Res. Thoughts (IJCRT), vol. 11, no. 3, pp. 414–421, Mar. 2023. [Online]. Available: https://ijcrt.org/papers/IJCRT2303747.pdf

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

[7] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques," Artif. Intell. Rev., vol. 26, no. 3, pp. 159–190, 2006.

[8] S. Rani, "Fraud detection in UPI using SMOTE, PCA, and XGBoost," Int. J. Comput. Appl., vol. 185, no. 10, pp. 23–29, 2023.

[9] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, Oct. 2001.

[10] V. N. Vapnik, The Nature of Statistical Learning Theory, 2nd ed. New York, NY, USA: Springer, 2000.

[11] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, USA, Aug. 2016, pp. 785–794.