Evaluating Machine Learning Based Behavioral Analytics for Detecting Unusual Transactional Sequences in High-Velocity UPI Digital Payment Networks

Mrs. Aarti Atul Pathade
Assistant Professor, MIT of Arts, Commerce and Computer Science, Alandi, Pune, 421162
doi.org/10.64643/IJIRT211387

***Abstract*—The Unified Payments Interface (UPI) allows fast bank-to-bank payments. This speed makes early fraud detection important. A payment may look normal on its own but become suspicious when compared with a user's recent activity. This paper proposes a simple way to study such patterns using machine learning. The system will use transaction details and summaries of earlier payments, such as the number of recent transfers, changes in spending, new payees and the time between payments. A rule-based method, Logistic Regression, Random Forest and Isolation Forest will be compared. Only one selected model will be used in the main scoring path. The proposed architecture has four steps: collect data, prepare behaviour features, calculate a risk score and flag suspicious payments for review. Testing will keep transactions in time order and measure fraud detection, false alerts and processing time. Public mobile-money data and clearly labelled synthetic UPI-like data will be used where suitable. Results from these sources will not be presented as proof of performance on live UPI payments. The study aims to find whether simple behaviour features can improve fraud detection without a complex system.**

***Index Terms*—UPI; fraud detection; machine learning; payment behaviour; unusual transaction patterns; false alerts; processing time.**

I. INTRODUCTION

1.1 Background

UPI is a payment system operated by the National Payments Corporation of India (NPCI). It supports instant transfers between bank accounts through participating applications [1]. Fast transfers leave a short period in which a payment provider can identify a suspicious payment. This paper does not describe all UPI transfers as non-refundable or impossible to reverse: NPCI procedures include transaction reversals, particularly for failed transactions [2]. Recovering money after a completed fraudulent transfer is a separate issue. A fraud check can consider both the current payment and the account's earlier activity. For example, several payments to new recipients within a short time may deserve closer review even when each amount is small. These are possible warning signs, not proof of fraud. A genuine user may also change devices, pay a new merchant or spend more than usual.

1.2 Problem statement

A system that checks only the current payment may miss patterns across several payments. At the same time, a system with many deep-learning and graph models can be difficult to build, test and explain within a small research project. The problem addressed here is whether simple summaries of earlier account activity can improve fraud detection while keeping processing time and false alerts at acceptable levels.

1.3 Scope and research focus

The proposed study focuses on account-level payment patterns, not on building a full banking security platform. It will compare a small group of methods under the same test conditions. It will not attempt to deploy Transformers, graph neural networks, federated learning or a two-stage deep-model pipeline. These remain possible future extensions. The main research idea is unchanged: compare a payment with earlier behaviour rather than judge it in isolation. The implementation scope is deliberately reduced.

II. AIM AND OBJECTIVES

Aim: To evaluate a simple and understandable machine-learning approach for identifying suspicious UPI-like payment patterns.

Objectives

- Review relevant work on payment fraud and behaviour-based features.
- Prepare transaction data and build features using only information available before the current payment decision.
- Compare a fixed-rule baseline, Logistic Regression, Random Forest and Isolation Forest.
- Measure fraud detection, false alerts, processing time and performance when payment behaviour changes.
- Develop a four-step prototype with clear alert reasons and a record of review outcomes.

III. LITERATURE REVIEW

3.1 Why transaction history matters

Bahnsen and colleagues studied feature engineering for credit-card fraud detection. Their work used aggregated transactions to describe spending behaviour and added features for recurring payment times [3]. This supports testing simple account-history features in the present study. However, credit-card evidence does not establish the same performance on UPI; the proposed experiments must test that transfer of ideas.

3.2 Synthetic mobile-money data

Lopez-Rojas, Elmir and Axelsson introduced PaySim, a simulator that generates mobile-money transactions for fraud-detection research [4]. Its simulation approach helps researchers work without exposing private transaction records. The public dataset is based on a mobile-money service rather than UPI [5]. It is therefore useful as a supporting benchmark, not as a substitute for genuine UPI data.

3.3 Relationship to the original review

The supplied draft discusses supervised models, anomaly detection, sequence models and graph models. Those topics provide a broader research background. This revision narrows the planned experiment to simple models with account-history features. It does not claim that these features reproduce the full abilities of an LSTM, Transformer or graph model. A small, reproducible comparison is the planned contribution, rather than a claim that the selected methods are new inventions.

3.4 Research gap to investigate

The study will investigate how much simple account-history features add under one consistent evaluation process. It will use the same time-based splits, feature rules and alert limits for each method. The review is preliminary, so the paper does not claim that no open UPI benchmark exists or that no earlier study has made a similar comparison. Any stronger novelty claim must follow a completed literature search.

Evidence or design issue	Meaning for this study
Behaviour features in card fraud [3]	Test history summaries, but do not assume the same benefit on UPI.
PaySim mobile-money simulation [4–5]	Use as a supporting dataset and report its domain limits.
Many models in the original proposal	Replace the broad implementation plan with four comparison methods.
Need for trustworthy results	Separate proposed outcomes from completed measurements.

IV. PROPOSED SIMPLE ARCHITECTURE

The architecture contains four steps. A single application and one database or local data store are enough for the first prototype. The system will replay transactions in time order; it is not connected to live banking infrastructure.

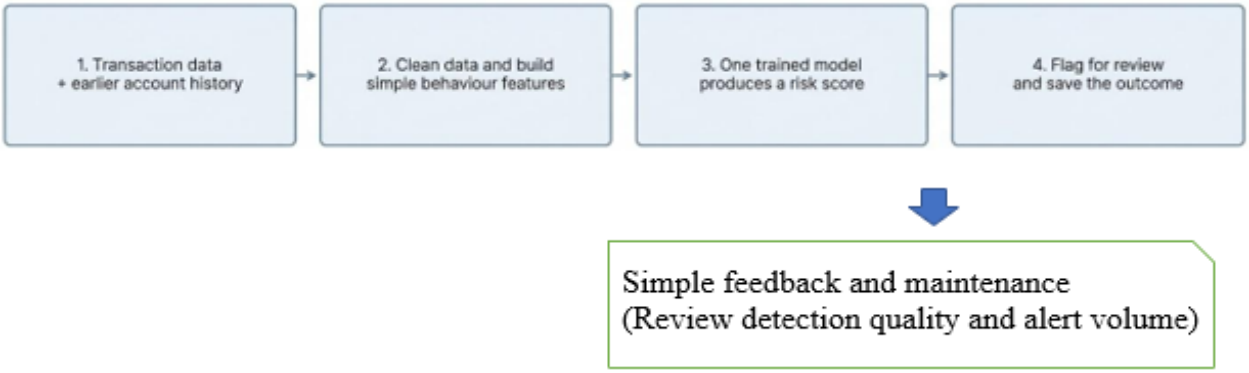
4.1 Step 1: Collect transaction data

Read transactions from an approved dataset or a synthetic-data file. Required fields are a transaction identifier, account identifier, time and amount. Recipient identifiers and transaction type will be used when available. Device or location fields are optional and must not be invented for a public dataset that lacks them. Keep the fraud label separate from the fields used to score a payment.

4.2 Step 2: Clean data and prepare behaviour features

Check duplicates, invalid amounts, missing fields and time order. For each account, maintain simple summaries of earlier transactions. Examples include payment counts over a recent window, the earlier

average amount, the time since the last payment and whether the recipient has been seen before. Calculate the current payment’s features before adding that payment to the account’s stored history. All learned cleaning and scaling settings come from training data only.



4.3 Step 3: Calculate a risk score

Compare the candidate methods during experiments, then select one model using validation results. The chosen model produces a risk score for each transaction. A rule baseline remains a separate comparison, not a mandatory first stage in a cascade. An Isolation Forest score is an unusualness score and must not be called a fraud probability. Other model scores will also be checked before being described as probabilities.

4.4 Step 4: Flag and record

Apply a threshold chosen on validation data. Payments above the threshold are marked for review; the prototype does not automatically block a real payment. Each alert records its score, model version, time and supporting observations, such as “new recipient” or “recent payment count above the account’s earlier level.” These observations explain the warning context but do not establish the cause of fraud. Store reviewer outcomes separately from confirmed fraud labels.

4.5 Simple feedback and maintenance

Review detection quality and alert volume on later time periods. Where enough confirmed labels exist, retrain offline and test the replacement model before use. Keep a record of model versions and threshold changes. No automatic model update or live banking decision is proposed at this stage.

V. RESEARCH METHODOLOGY

The proposed research framework systematically evaluates user-centric payment behaviors to detect UPI transaction anomalies through seven simplified sequential phases:

5.1. Literature Search

5.1.1.Databases:

Academic literature searches will be strictly conducted across Google Scholar, IEEE Xplore, and ACM Digital Library.

5.1.2.Keywords:

Search queries will utilize unified string combinations of "UPI", "Fraud Detection", "Machine Learning", and "Behavior Features". Duplicates and non-peer-reviewed preprints will be recorded and excluded.

5.2 Data Sources and Limits

The experimental infrastructure relies entirely on unlabelled and synthetic data streams, removing dependencies on restricted institutional bank data:

5.2.1.PaySim Benchmark:

A standard, open-source mobile money simulation dataset used to establish operational baseline parameters.

5.2.2.Synthetic UPI-like Scenarios:

Custom generated transaction datasets modeling normal transactional behavior alongside simulated exceptional scenarios (such as flash commerce rushes, sudden geographic telemetry hops, and high-velocity velocity anomalies).

5.3 Behavioral Feature Plan

Transactions will be transformed into user-behavior profiles using only historical parameters known before the switch validation window closes:

5.3.1. Current Metric:
Transaction amount and exact execution timestamp.

5.3.2. Recent Activity:
Rolling historical counts and cumulative transaction amounts over time windows matching the data resolution.

5.3.3. Velocity Shifts:
Current transaction cost divided by individual moving averages.

5.3.4.Contextual Features:
Flags indicating temporary device shifts or interaction with previously unrecorded beneficiaries.

5.4 Algorithmic Models for Comparison
Four computational methods will be cross-evaluated to compare classical threshold structures against adaptive predictive intelligence:

5.4.1.Fixed Rules:
A transparent baseline operating on rigid validation-tuned transactional limits.

5.4.2.Logistic Regression:
A lightweight linear supervised baseline model.

5.4.3.Random Forest:
An ensemble supervised learner capable of mapping non-linear behavioral indicators across tabular transaction blocks.

5.4.4.Isolation Forest:
An unsupervised anomaly detector deployed to isolate zero-day pattern deviations when explicitly labeled training records are absent.

5.5 Training, Validation, and Testing Split
To preserve experimental integrity and prevent look-ahead bias, datasets will be partitioned strictly in chronological order:

5.5.1.Training Block (60%):
Used exclusively to fit preprocessing parameters and train model weights.

5.5.2.Validation Block (20%):
Used to fine-tune alert thresholds, adjust hyper-parameters, and calibrate data-balancing weights (e.g., class-weighted cost metrics).

5.5.3.Testing Block (20%):
Frozen until the final runtime phase to compute unbiased accuracy metrics.

5.6 Evaluation Metrics
Given the extreme class asymmetry where legitimate UPI traffic represents over 99.9% of streaming inputs, overall accuracy is rejected as an evaluation tool. Instead, models are benchmarked using:

5.6.1.Precision and Recall:
Maximizing fraud caught while minimizing false alerts.

5.6.2.F1-Score and AUPRC:
Measuring the geometric balance across risk score boundaries.

5.6.3.Computational Footprint:
Recording the median and 95th-percentile inference latency (measured in milliseconds) alongside CPU and memory consumption.

5.7 Privacy and Responsible Use
The experimental prototype uses no Personally Identifiable Information (PII). All user and bank accounts are simulated as randomized token identifiers. The final output is optimized strictly for offline research evaluation and holds no capability for active pipeline intervention without external regulatory, legal, and engineering compliance tracking.

VI. PROPOSED HYPOTHESES

H1: Incorporating earlier-behaviour features into a supervised machine-learning model improves the recall of unusual transactional sequences in high-velocity UPI payment networks at a fixed alert budget, compared with models using only current-transaction features.

VII. EXPECTED OUTCOMES AND CONTRIBUTIONS

The intended outputs are a reproducible feature-building process, a fair comparison of a small set of methods, and a working four-step prototype. The study may show that simple history summaries improve

detection, but the result is not guaranteed. Its contribution will be the evidence from the comparison and the clarity of the evaluation, not the invention of a new machine-learning algorithm. Any recommendation will be limited to the datasets and test conditions used.

VIII. LIMITATIONS

- Synthetic UPI-like data may reflect the generator’s assumptions more than actual user behaviour.
- PaySim represents mobile money and cannot establish performance on UPI by itself.
- Historical summaries may miss long or coordinated multi-account fraud patterns.
- An unusual payment is not necessarily fraudulent; social-engineering payments may resemble normal activity.
- Delayed or incorrect labels can affect both training and evaluation.
- A local replay experiment cannot establish production readiness, bank-scale capacity or compliance.

IX. CONCLUSION

This proposal keeps the focus on unusual payment patterns while reducing the complexity of the planned system. It compares a small group of methods using current-payment details and simple summaries of earlier account activity.

The four-step architecture collects data, prepares features, calculates a score and flags suspicious payments for review. Testing will keep transactions in time order and report detection quality, false alerts and processing time. No experimental success is claimed at this stage. Further work with authorised UPI data would be needed before applying the approach to live payments.

REFERENCES

[1] A. Khedekar and G. Manerkar, “Machine learning-driven detection of UPI fraudulent activities,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 12, no. 4, 2025.

[2] T. S. Kartik, G. C. Teja, Williams, and K. Raghu, “Adaptive UPI fraud detection: A hybrid machine learning framework,” in *Proc. ICCSCE*, 2025, doi: 10.2991/978-94-6463-858-5_51.

[3] B. Sethi, S. Mhatre, S. Yadav, S. Das, and V. Jadhav, “Machine learning-based UPI fraud detection: A comprehensive approach using Random Forest,” in *Proc. ICAIEHS*, 2025.

[4] P. Sirisha et al., “Unified Payments Interface fraud detection using machine learning,” in *Proc. ICRDICCT*, 2025, pp. 420–426, doi: 10.5220/0013930800004919.

[5] M. Vijaya Sree et al., “Enhanced detection of UPI frauds using advanced machine learning techniques for secure digital transactions,” in *Proc. ICITSM*, 2025, doi: 10.4108/eai.28-4-2025.2357830.

[6] A. Abdallah, M. A. Maarof, and A. Zainal, “Fraud detection system: A survey,” *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, 2016.

[7] M. Ahmed, A. N. Mahmood, and M. R. Islam, “A survey of anomaly detection techniques in financial domain,” *Future Generation Computer Systems*, vol. 55, pp. 278–288, 2016.

[8] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, “Feature engineering strategies for credit card fraud detection,” *Expert Systems with Applications*, vol. 51, pp. 134–142, 2016.

[9] R. J. Bolton and D. J. Hand, “Statistical fraud detection: A review,” *Statistical Science*, vol. 17, no. 3, pp. 235–255, 2002.

[10] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: Identifying density-based local outliers,” in *Proc. ACM SIGMOD Int. Conf. Management of Data*, 2000, pp. 93–104.

[11] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, “Combining unsupervised and supervised learning in credit card fraud detection,” *Information Sciences*, vol. 557, pp. 317–331, 2021.

[12] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” arXiv preprint arXiv:1901.03407, 2019.

[13] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[14] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[15] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, “Learned lessons in credit card fraud detection from a practitioner perspective,” *Expert Systems with Applications*, vol. 41, no. 10, pp. 4915–4928, 2014.

[16] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, “Calibrating probability with undersampling for unbalanced classification,” in *Proc. IEEE Symp. Series on Computational Intelligence (SSCI)*, 2015, pp. 159–166.

[17] M. Du, F. Li, G. Zheng, and V. Srikumar, “DeepLog: Anomaly detection and diagnosis from system logs through deep learning,” in *Proc. ACM SIGSAC Conf. Computer and Communications Security*, 2017, pp. 1285–1298.

[18] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, “Using generative adversarial networks for improving classification effectiveness in credit card fraud detection,” *Information Sciences*, vol. 479, pp. 448–455, 2019.

[19] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 1024–1034.

[20] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, and O. Caelen, “Sequence classification for credit-card fraud detection,” *Expert Systems with Applications*, vol. 100, pp. 234–245, 2018.

[22] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. Int. Conf. Learning Representations (ICLR)*, 2017.

[23] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422.

[24] E. Lopez-Rojas, A. Elmir, and S. Axelsson, “PaySim: A financial mobile money simulator for fraud detection,” in *Proc. 28th European Modeling and Simulation Symposium*, 2016.

[25] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4765–4774.

[26] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, “Long short-term memory networks for anomaly detection in time series,” in *Proc. European Symposium on Artificial Neural Networks (ESANN)*, 2015, pp. 89–94.

[27] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, “LSTM-based encoder-decoder for multi-sensor anomaly detection,” arXiv preprint arXiv:1607.00148, 2016.

[28] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, “The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature,” *Decision Support Systems*, vol. 50, no. 3, pp. 559–569, 2011.

[29] G. Pang, C. Shen, L. Cao, and A. van den Hengel, “Deep learning for anomaly detection: A review,” *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, 2021.

[30] C. Phua, V. Lee, K. Smith, and R. Gayler, “A comprehensive survey of data mining-based fraud detection research,” arXiv preprint arXiv:1009.6119, 2010.

[31] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature Machine Intelligence*, vol. 1, pp. 206–215, 2019.

[32] L. Ruff et al., “Deep one-class classification,” in *Proc. Int. Conf. Machine Learning (ICML)*, 2018, pp. 4393–4402.

[33] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” in *Proc. Int. Conf. Information Processing in Medical Imaging (IPMI)*, 2017, pp. 146–157.

[34] A. Vaswani et al., “Attention is all you need,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[35] M. Weber et al., “Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics,” in *Proc. KDD Workshop on Anomaly Detection in Finance*, 2019.

[36] B. Zong et al., “Deep autoencoding Gaussian mixture model for unsupervised anomaly detection,” in *Proc. Int. Conf. Learning Representations (ICLR)*, 2018.

[37] National Payments Corporation of India, “UPI product statistics and circulars,” [Online]. Available: https://www.npci.org.in/. [Accessed: Nov. 20, 2026].

[38] Reserve Bank of India, “Master directions and guidance on digital payment security controls and fraud risk management,” [Online]. Available: https://www.rbi.org.in/. [Accessed: Nov. 20, 2026].