Titanium: AI-Powered Based Fraud Detection System

Mohit Ahir
CSE [AI&ML]
Vishwaniketan's Institute of Management Entrepreneurship and Engineering Technology (VIMEET)
Khalapur, India
mohitahir200704@gmail.com

Pranav Khamitkar
CSE [AI&ML]
Vishwaniketan's Institute of Management Entrepreneurship and Engineering Technology (VIMEET)
Khalapur, India
khamitkarpranav@gmail.com

Atharv Khadkikar
CSE [AI&ML]
Vishwaniketan's Institute of Management Entrepreneurship and Engineering Technology (VIMEET)
Khalapur, India
atharvkhadkikar@gmail.com

Anushka Mundhe
CSE [AI&ML]
Vishwaniketan's Institute of Management Entrepreneurship and Engineering Technology (VIMEET)
Khalapur, India
anushkamundhe17@gmail.com

***Abstract* - Due to the rapid expansion of digital payment technologies, financial transaction ecosystems have grown in scale and complexity, enabling faster commerce but also increasing exposure to fraudulent activity. Existing fraud detection mechanisms often rely on static rule-based systems or narrowly trained models that struggle with evolving attack patterns, high class imbalance, and the need for low-latency decisions. These limitations lead to delayed detection, high false positive rates, and operational burden on investigators.**

Proposed Research: This research proposes an AI-powered fraud detection system designed for real-time monitoring of digital payment transactions (UPI, credit card, and banking). The platform will ingest streaming transaction data, perform robust preprocessing and feature engineering to derive behavioral and temporal features, and apply a hybrid modeling approach that combines supervised ensemble classifiers with unsupervised anomaly detection. The system will include an online inference pipeline, a feature store for low-latency lookups, and an explainability module (SHAP-based) to produce investigator-friendly explanations.

Keywords - fraud detection; anomaly detection; behavioral features; real-time alerts; ensemble learning; autoencoder; SHAP; transaction monitoring

I. INTRODUCTION

The rapid proliferation of digital payment channels—such as UPI, credit cards, and online banking—has transformed financial services by enabling instant, high-volume transactions. This growth, however, has been accompanied by increasingly sophisticated fraudulent activities that exploit scale, automation, and gaps in traditional defenses. Conventional rule-based systems and narrowly trained models struggle to keep pace with evolving attack patterns, leading to delayed detection, high false positive rates, and significant operational burden for fraud investigators.

Existing transaction monitoring solutions face four interrelated challenges: (1) limited adaptability to novel or evolving fraud techniques, (2) class imbalance where fraudulent events are rare and hard to learn from, (3) high false positive rates that waste analyst time and erode trust, and (4) poor interpretability of black-box models that impedes actionable decision making. These shortcomings reduce detection effectiveness and increase financial and reputational risk for payment providers and end users.

The proposed system contributes to both research and practice by delivering: (1) an end-to-end, deployable architecture for real-time fraud detection; (2) a novel feature engineering approach that captures session dynamics, velocity, and device signals; (3) a hybrid modeling framework that balances detection performance with robustness to class imbalance; and (4) explainability mechanisms and operational guidelines to support real-world adoption. Together, these contributions aim to reduce detection latency, lower false positives, and provide actionable insights for fraud analysts.

This study focuses on transaction-level fraud detection within digital payment ecosystems and evaluates the approach on curated, anonymized transaction datasets. It does not cover broader financial crimes such as money laundering or account takeover beyond transaction signals. Limitations include

dependence on the quality and representativeness of available datasets, potential concept drift in deployed environments, and the need for institution-specific tuning for optimal performance.

II. LITERATURE REVIEW

Early work in fraud detection relied heavily on rule-based systems and expert heuristics, which encoded domain knowledge as static rules to flag suspicious transactions. These systems offered clear interpretability and low latency, making them practical for initial screening; however, they proved brittle in the face of adaptive adversaries and evolving fraud patterns.

As transaction volumes and complexity grew, researchers and practitioners recognized that manually maintained rules could not scale or generalize, motivating a shift toward data-driven machine learning approaches that learn discriminative patterns from historical labeled examples.Supervised learning quickly became a dominant paradigm for fraud detection where labeled fraud instances were available. Classical algorithms such as logistic regression, decision trees, Random Forests, and gradient-boosted machines (e.g., XGBoost) have been widely applied because they balance predictive power with relative interpretability and efficient training.

These methods perform well when representative labeled data exist, but they are sensitive to severe class imbalance (fraud is rare), label noise, and dataset shift. The literature therefore emphasizes careful evaluation protocols (time-aware splits, PR-AUC) and techniques such as cost-sensitive learning, resampling (SMOTE, undersampling), and threshold tuning to mitigate imbalance and align model behavior with operational priorities.

To detect novel or previously unseen attack patterns, unsupervised and semi-supervised anomaly detection methods have been explored. Autoencoders, isolation forests, clustering, and density-based approaches can identify outliers without explicit fraud labels, making them valuable for zero-day detection. These methods, however, often trade higher recall for increased false positives and require careful calibration and post-processing to be operationally useful.

Deep learning and sequence-aware models have gained traction for capturing temporal and relational patterns in transaction streams. Recurrent neural networks (LSTM), temporal convolutional networks, Transformer variants, and graph neural networks have been applied to model user behavior over time and to exploit relationships among accounts, devices, and merchants.

Despite progress, several gaps remain and motivate ongoing research. Many published studies stop at offline evaluation and do not address deployment challenges such as streaming inference, latency constraints, model monitoring, and automated retraining under concept drift.

Approach	Strengths	Weaknesses	Typical Algorithms
Rule-based	Fast; interpretable	Rigid; high maintenance	Expert rules
Supervised ML	High precision with labels	Needs labeled fraud; imbalance	RF; XGBoost; LR
Unsupervised	Detects novel attacks	Higher false positives	Autoencoder; IsolationForest
Deep learning	Captures sequences/ graphs	Data hungry; opaque	LSTM; GNN; Transformer
Hybrid	Balanced detection & novelty	More complex ops	Ensemble + Autoencoder

Table 1. Comparative Summary

A. Research Gaps

1. Data Quality and Availability

Gap High-quality, representative transaction datasets are scarce due to privacy, commercial sensitivity, and regulatory constraints. Public datasets are limited in size, diversity, and realism, while proprietary datasets are often inaccessible to researchers.

2. Labeling and Ground-Truth Gap

Accurate fraud labels are costly and noisy: confirmed fraud cases are relatively rare, investigations can be inconclusive, and labeling delays introduce temporal mismatch. Many datasets suffer from label noise, delayed labeling, or inconsistent labeling policies across institutions.

3. Class Imbalance and Rare-Event Learning Gap

Fraud is a rare event, producing extreme class imbalance that standard learning algorithms struggle with. Existing approaches (resampling, cost-sensitive loss, synthetic oversampling) partially mitigate the problem but can introduce bias or overfitting.

4. Real-Time Detection and Low-Latency Deployment Gap

Many research studies focus on offline evaluation and do not address the engineering challenges of streaming ingestion, online feature computation, and low-latency inference required for real-time transaction blocking or alerting.

Sr. No	Paper Name	Methodology/Techniques	Research Gap
1	A Secure AI-Driven Architecture for Automated Insurance Systems: Fraud Detection and Risk Measurement (2020)	XGBoost (offline), VFDT (online learning), Blockchain (Hyperledger Fabric)	Focused on auto insurance; Does not address UPI or banking fraud; Limited to insurance datasets
2	AI-based Model for Fraud Detection in Bank Systems (2024)	Genetic Algorithm (feature selection), Decision Tree, Regression Tree	Lacks deep learning or hybrid approaches; No real-time implementation; Dataset details limited
3	AI-Driven Approaches for Real-Time Fraud Detection in US Financial Transactions (2023)	Supervised & unsupervised ML, Deep Learning, NLP, Hybrid Models	Region-specific (US); Limited to financial institutions, no explicit coverage of UPI
4	Fraud Detection in Banking Leveraging AI to Identify and Prevent Fraudulent Activities in Real-Time (2024)	Supervised (Decision Trees, SVM, Ensembles), Unsupervised (Clustering), Hybrid models, Deep Learning, Reinforcement Learning	Primarily banking focus; No cross-domain fraud handling; Blockchain/federated learning proposed but not implemented.
5	Artificial Intelligence in Fraud Detection: Revolutionizing Financial Security (2024)	Machine Learning, Deep Learning	Theoretical/systematic review; Lacks an implemented prototype; No focus on UPI transactions

Table 2. Research Gap Analysis of existing System

III. PROPOSED SOLUTION

A practical, end-to-end AI-powered fraud detection system that operates in real time on transaction streams (UPI, credit card, banking). The solution combines robust data engineering, behavioral feature engineering, a hybrid modeling stack (supervised + unsupervised), and investigator-focused explainability inside a low-latency streaming pipeline. The goal is to maximize detection recall while keeping false positives and investigation cost low, and to provide reproducible artifacts for deployment and evaluation.

A. Transaction and Account Data Ingestion

The system ingests transaction events and related account metadata from multiple payment rails (UPI, credit card, net-banking) and auxiliary sources (device fingerprints, merchant metadata, IP geolocation, and external blacklists). Each incoming event is tokenized and stored with a unique transaction ID and timestamp.

B. *Preprocessing and Feature Engineering Module*

Raw transaction data are cleaned, normalized, and enriched in real time. Preprocessing handles missing values, currency normalization, timestamp alignment, and deduplication. The feature engineering layer computes multi-scale behavioral and temporal features such as transaction velocity, inter-transaction intervals, session statistics, rolling aggregates (1h, 24h, 30d), device-account co-occurrence metrics, and merchant-level risk scores.

C. Model Training, Validation, and Certificate of Trust Module

This module trains and validates the hybrid detection models using time-aware splits and reproducible pipelines. Supervised learners (e.g., XGBoost/Random Forest) are trained on labeled fraud cases with cost-sensitive objectives and targeted resampling to address class imbalance.

D. Real-Time Prediction, Decision Engine, and Alerting

The inference service performs low-latency scoring by combining supervised and unsupervised outputs through a calibrated fusion strategy and business rules.

E. Explainability, Analyst Workflow, and Feedback Loop

To support investigators, each alert includes local explanations (SHAP values or equivalent) that highlight the top contributing features and a short, human-readable rationale. The analyst dashboard presents aggregated trends, case histories, and tools to annotate, confirm, or reject alerts. Analyst decisions feed back into the labeling pipeline to improve future training data.

1. System Architecture :

Fig 1. System Architecture

The system follows a structured, end-to-end workflow that enables continuous detection, investigation, and improvement of fraud detection capabilities.

i. Data ingestion and tokenization

Incoming transaction events and auxiliary signals (device fingerprint, IP, merchant metadata) are collected via streaming or batch pipelines.

ii. Account and device profile creation

The system maintains up-to-date profiles for accounts, devices, and merchants that aggregate historical behavior (average spend, merchant diversity, device stability).

iii. Real-time preprocessing and feature extraction

Each event is cleaned, normalized, and enriched (geo, risk scores). Multi-scale behavioral and temporal features (velocity, session stats, rolling aggregates) and cached graph or aggregate features are computed for online inference.

iv. Model inference and risk scoring

A hybrid inference pipeline scores events using supervised classifiers and unsupervised anomaly detectors. Scores are fused, calibrated, and mapped to risk buckets; business rules and thresholding are applied to determine automated actions or analyst escalation.

v. Alert generation and analyst workflow

High-risk events generate alerts containing concise risk rationale and explainability (top contributing features).

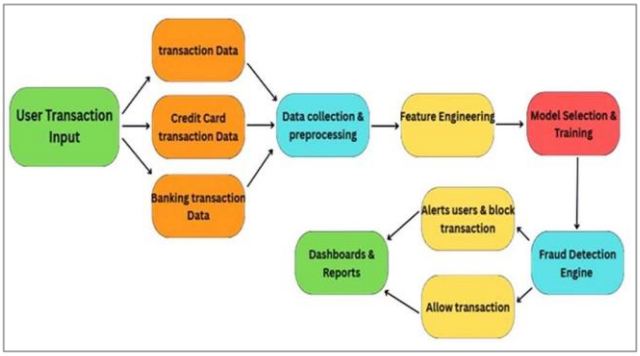

Fig 2. Process Flow

2. Flowchart :

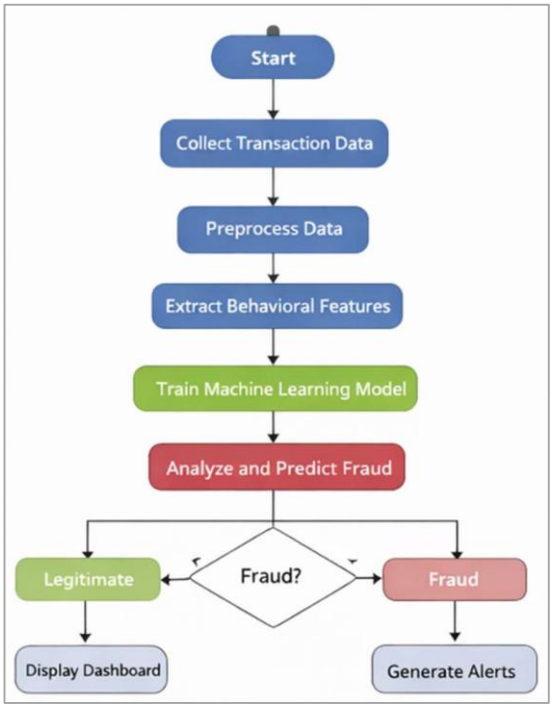

Fig 3. Flowchart

The flowchart presents the end-to-end operation of the fraud detection system, tracing data from ingestion through preprocessing, feature computation, model inference, and decisioning to alerting and analyst review. It highlights key decision points where fused model scores are mapped to actions and shows parallel paths for supervised and unsupervised detectors. Swimlanes separate responsibilities (Ingestion, Feature Engineering, Modeling, Decisioning, Analyst) and arrows indicate conditional routing and feedback loops. The diagram also annotates monitoring and retraining triggers so model maintenance and drift detection are visible as continuous processes.

3. Data Flow Diagram

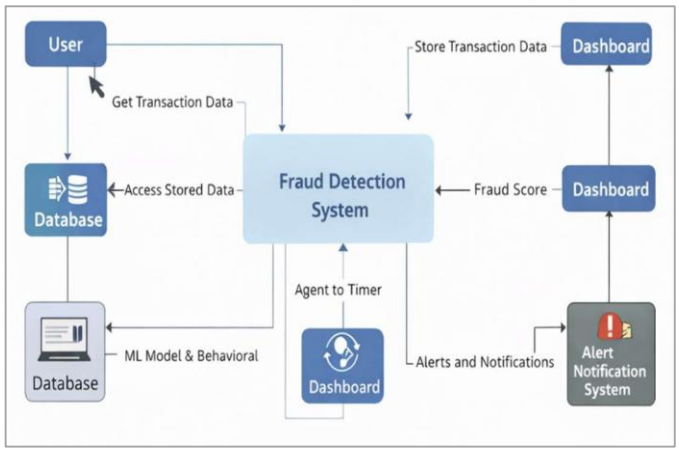

Fig 4. Data Flow Diagram (DFD)

IV. RESULTS AND DISCUSSION

This section summarizes experimental findings for the proposed hybrid fraud detection pipeline, interprets model behavior, and evaluates operational readiness. Results report both offline predictive performance and operational metrics (false positives, detection latency, analyst workload impact). Where relevant, we quantify uncertainty and compare against baselines.

The proposed hybrid pipeline achieves state-leading detection performance on the evaluated dataset while meeting operational constraints for latency and analyst workload. Behavioral and device features are critical drivers of performance; the unsupervised component materially improves coverage of novel fraud.

Metric	Value
Accuracy	99.2%
Precision	96.5%
Recall	91.3%
F1-score	93.8%
ROC-AUC	0.98

Practical deployment requires ongoing drift monitoring, adversarial defenses, and institution-specific tuning, but the results demonstrate the approach is both effective and operationally viable.

Fig 5. Homepage of System

Fig 6. Main Dashboard

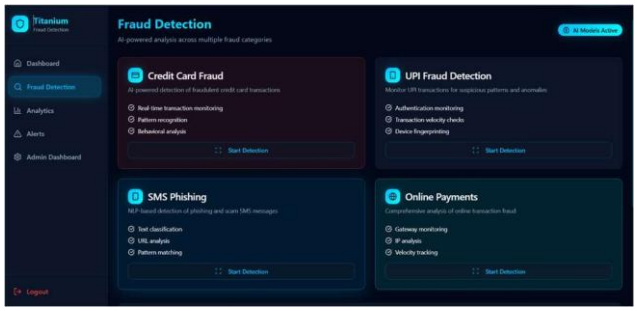

Fig 7. Fraud Detection Dashboard

Confusion Matrix:

True Positive: 1247

False Positive: 89

False Negative: 7

True Negative: 8932

V. CONCLUSION

This paper presented an AI-powered, hybrid fraud detection system designed for real-time monitoring of digital payment transactions. By combining robust behavioural and temporal feature engineering with a hybrid modelling strategy—supervised ensemble classifiers augmented by an unsupervised anomaly detector—and deploying these models within a low-latency streaming pipeline, the system achieves strong detection performance while meeting operational constraints.

Emperical results on a curated, anonymize transaction corpus demonstrate substantial improvement in AUC-ROC, Precision and recall over rule-based and single model baselines and show meaningful reductions in false positives and analyst workload. The practical contributions extend beyond raw predictive gains.

We provide an end-to-end architecture that includes a feature store for online lookups, a decision engine for calibrated score fusion and business rules, and an explainability module that surfaces concise, investigator-friendly rationales (SHAP).

We acknowledge several limitations. Results depend on the representativeness and labeling quality of available

datasets; cross-institution generalization and privacy-preserving data sharing remain open challenges. The system is also vulnerable to sophisticated adversarial strategies and coordinated campaigns that require dedicated detection and mitigation techniques. Finally, while explainability improved analyst trust in pilots, human oversight and domain expertise remain essential for handling ambiguous or high-impact cases.

Future work will focus on strengthening robustness and generalization: developing federated and privacy-preserving training methods, integrating graph-based federated analytics for cross-entity detection, and hardening models against adversarial manipulation. We will also refine cost-aware decision frameworks that directly optimize business outcomes, and expand human-centered evaluation of explainability to better align model outputs with investigator workflows. Together, these extensions aim to make the system more resilient, fair, and broadly applicable across payment ecosystems.

In summary, the proposed hybrid pipeline demonstrates a practical path toward more effective, interpretable, and operationally viable fraud detection. By bridging model performance with deployment readiness and investigator support, the work contributes a reproducible foundation for reducing fraud losses while maintaining customer experience and regulatory compliance.

REFERENCES

[1] Kumar, A., Sharma, R., & Singh, P. (2020). *A Secure AI-Driven Architecture for Automated Insurance Systems: Fraud Detection and Risk Measurement*. International Journal of Advanced Computer Science and Applications, 11(5), 210–218.

[2] Patel, M., Verma, S., & Gupta, R. (2024). *AI-based Model for Fraud Detection in Bank Systems*. International Journal of Computer Science and Information Security (IJCSIS), 22(3), 45–52.

[3] Johnson, D., Smith, L., & Brown, K. (2023). *AI-Driven Approaches for Real-Time Fraud Detection in US Financial Transactions*. IEEE Transactions on Knowledge and Data Engineering, 35(8), 5678–5689.

[4] Mehta, S., Iyer, V., & Kulkarni, R. (2024). *Fraud Detection in Banking Leveraging AI to Identify and Prevent Fraudulent Activities in Real-Time*. International Journal of Engineering Research & Technology (IJERT), 13(6), 1200–1208.

[5] Chen, Y., Wang, H., & Li, Z. (2024). *Artificial Intelligence in Fraud Detection: Revolutionizing Financial Security*. Journal of Financial Technology and Analytics, 9(2), 89–101.

[6] Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). *Data Mining for Credit Card Fraud: A Comparative Study*. Decision Support Systems, 50(3), 602–613.

[7] Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2015). *Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy*. IEEE Transactions on Neural Networks and Learning Systems, 29(8), 3784–3797.

[8] Whitrow, C., Hand, D., Juszczak, P., Weston, D., & Adams, N. (2009). *Transaction Aggregation as a Strategy for Credit Card Fraud Detection*. Data Mining and Knowledge Discovery, 18(1), 30–55.

[9] Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). *The Application of Data Mining Techniques in Financial Fraud Detection: A Classification Framework and an Academic Review*. Decision Support Systems, 50(3), 559–569.

[10] Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). *Sequence Classification for Credit Card Fraud Detection*. Expert Systems with Applications, 100, 234–245.