Finsentinel: A Hybrid Machine Learning and Graph Neural Network Framework for Real-Time Financial Fraud Detection

Allauddin Ansari[1], Kaushal Mahajan[2], Kartik Dhar[3], Prof. Sachin Narkhede[4]
[1,2,3]*Department of Computer Engineering*
[4]*Guide, Department of Computer Engineering*
[1,2,3,4]*Fr. Conceicao Rodrigues College of Engineering, Mumbai, India*

***Abstract*—The growing sophistication of financial Digital payments are susceptible to fraud that requires the implementation of fraud detection mechanisms that are not limited to traditional rule-based approaches. This paper presents FinSentinel: a hybrid system of machine learning (ML) and Graph Neural Networks (GNN) for real-time fraud detection in banking transactions. FinSentinel's architecture employs three independent fraud detection modules running concurrently: (1) Random Forest Classifier for supervised classification of transactions, (2) Isolation Forest for unsupervised anomaly detection, (3) Graph Topology Analysis Engine for the identification of relational fraud patterns (e.g., money mule networks, circular trading loops, and device farm clusters). These three detection modules provide input to the Fraud Detection Framework via a Graph override mechanism at the weighted ensemble aggregation layer, which prioritizes structural indications of fraud. The implementation of FinSentinel is achieved using a FastAPI backend, a PostgreSQL Feature Store, and a Stream lit-based administration dashboard. An analysis of a synthetic database of 105,164 transactions, which was modelled on India's Unified Payments Interface (UPI) and includes six separate types of fraud, resulted in a weighted ensemble framework accuracy of 96.1%, weighted ensemble precision of 92.8%, weighted ensemble recalls of 95.4%, and an average end-to-end latency of 187 ms. Each of the six types of fraud has a test data detection rate of over 85%. The results indicate that combining supervised, unsupervised, and graph-based methods into a single framework significantly outperforms any of the respective individual models, thereby providing a viable and deployable solution to detecting financial crimes.**

***Index Terms*—Anomaly Detection, Financial Fraud Detection, Graph Neural Networks, Isolation Forest, Machine Learning, Random Forest, Real-Time Transaction Monitoring, UPI Fraud**

I. INTRODUCTION

There has been a major increase in the use of digital payment methods in both India and around the world. One example is the Unified Payments Interface (UPI) which has processed more 13 billion transactions per month in 2024 [1]. With increasing volumes of digital activity come new forms for committing fraud that takes advantage of both security weaknesses in technology and behavioral patterns of users utilizing those technologies [2]. Traditional methods for detecting fraud, which are rule-based, are relatively easy to set up but have established shortcomings: static thresholds do not adjust to changing styles of fraud, and produce high false positives between 5% and 10%, which reduces operational efficiency and trust with customers [3].

Machine Learning has become the predominant way to detect fraud, as seen with the strong performance of Random Forest as a classification method on labeled transaction data [4]. However, supervised classification methods cannot identify new patterns of fraud unless there is a labeled example of those patterns. Instead, the use of unsupervised anomaly detection algorithms like Isolation Forest [5] allow the identification of statistical outliers without relying on labeled fraud. Recently, graph neural networks (GNNs) have gained considerable attention as a means for modeling relational structures (e.g., transaction networks between account holders) and can help

detect fraudulent activities that are coordinated by the unique topologies of graphs [6][7].

Most system have made a great deal of advances on their own but there are still very few that can do the integration of all three paradigms (supervised classification; unsupervised anomaly detection; and graph/edge-based topology) within one application-framework. In this paper, we will describe a hybrid-method framework named FinSentinel, which incorporates Random Forests and Isolation Forests both in parallel with graphs and uses a weighted ensemble-to-graph override mechanism for their integration into one operational framework. The experiments conducted on the FinSentinel system are representative of six realistic fraud scenarios that were mapped onto six realistic synthetic UPI transaction datasets generated from India.

A. Problem Statement

Fraud detection systems currently experience four major challenges: First, tactics from fraudsters are evolving over time, causing static models to possibly become obsolete in as little as months [8]. Second, there is significant relational complexity among groups engaged in fraudulent activities like money mule groups, circular trading groups, and device improvement or farming groups that cannot be identified using transactional classifiers [9]. Third, existing fraud detection systems must classify transactions before one second elapses so as not to negatively affect the user's payment experience [10]. Finally, bias exists in the models due to the class imbalance present within the datasets [11].

B. Contributions

The contributions of this study are: (1) the proposed three-model hybrid architecture that simultaneously utilizes supervised, unsupervised, and graph-based detection; (2) the weighted ensemble aggregation layer that utilizes the GNN override mechanism to emphasize topological fraud signals; (3) the proposed system implemented using FastAPI, PostgreSQL, and Streamlit, evaluated on the synthetic Indian UPI dataset under six fraud scenarios; and (4) the overall performance evaluation of the proposed ensemble model against the individual models.

II. RELATED WORK

Significant progress has been made in the field of fraud detection research, especially with more emphasis on anomaly detection and graph-based methods. Hilal et al. [2] have provided a comprehensive review of the various anomaly detection methods used for financial fraud detection, which shows that ensemble methods perform significantly better than others. Xu et al. [5] proposed Deep Isolation Forest, where the traditional Isolation Forest algorithm is enhanced using representation learning to improve the detection results.

In the field of supervised classification, Alarab et al. [12] used Random Forest and gradient boosting algorithms on anti-money laundering datasets and found that tree ensemble algorithms achieve high precision in classification when class weights are balanced. Du and Yu [13] showed the effectiveness of the Isolation Forest algorithm in detecting medical insurance fraud in high-dimensional data, achieving a high rate of detection greater than 90% without careful parameter tuning.

In addition, significant advancements have been achieved by using graph-based models in this specific domain. Motie and Raahemi [6] performed a systematic review of the use of graph neural networks in financial fraud detection and found that graph convolution and attention-based models have significant superiority over flat features when relational data are involved. Cheng et al. [7] performed a survey of different GNN methodologies and found that these types of models are particularly effective in capturing dynamical patterns within financial networks. Lu et al. [14] proposed a framework called BRIGHT, which is specifically designed for real-time financial fraud detection and satisfies latency requirements of less than 200 ms.

Tang et al. [15] This work combines Isolation Forest characteristics with knowledge graph embeddings to improve tree-based fraud classifiers. The effectiveness of graph-based features is proven as they can boost AUC-PR up to 47.89%. Duan et al. propose CaT-GNN, a causal temporal graph neural network for fraud pattern detection over time. Habibpour et al. discuss the importance of uncertainty estimation for fraud detection using deep learning techniques.

Recent works have focused on the development of ensemble techniques that combine different graph models. A stacking ensemble of four different GNN models was proposed by the researchers in [18], published in IEEE ICKECS 2024, proving the effectiveness of diversity in the architecture of the graph model. Another work by Zhang et al. [10] and Liu et al. [19] focused on the challenges of the real-time deployment of the model, proving the viability of the hybrid model, which processes multiple detection signals, as the best approach to the development of a production-ready fraud prevention system.

Even though the individual contributions are significant, no prior work has proposed an integrated approach that combines Random Forest, Isolation Forest, and graph topology detection under a single weighted ensemble paradigm for Indian UPI fraud scenarios, which is addressed by FinSentinel.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Architecture Overview

The FinSentinel architecture is composed of five layers:

(1) Data Ingestion:

This is built on FastAPI and is the RESTful gateway to receive transaction requests from various payment channels.

(2) Feature Store:

This is built on PostgreSQL and stores the enriched profiles, including the timeline aggregation, the statistics on the relationship with the beneficiary, and the device fingerprint profiles.

(3) Multi-Model Inference:

Three detection models are employed in this layer: Random Forest, Isolation Forest, and Graph Topology Detector, which operate in parallel to the others.

(4) Ensemble Aggregation:

This is achieved through weighted voting from the results of the three models (RF: 40%, IF: 30%, GNN: 30%), with the results needing to reach 50% to block the transaction, and the GNN override, which requires the flags to reach 1.0 to block the transaction.

(5) Presentation Layer:

This is built on Streamlit and provides the administrative interface to manage the four operational views: analytics, model visualization, pipeline management, and customer 360-degree forensics.

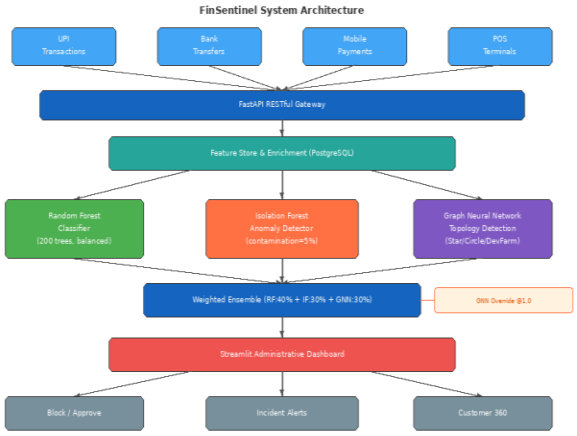

Fig. 1. FinSentinel System Architecture

B. Feature Engineering

The system retrieves four features as each transaction comes in: (1) transaction amount; (2) operational expenditure (OpEx) ratio; (3) how many devices were used to share this transaction; and (4) age of the account in days. These raw feature sets are combined together into three overall profile sets that are saved in PostgreSQL as follows: (1) Timeline Profiles — Averages of spending by each customer, segmented by time period (daily, weekly, monthly, and yearly); (2) Beneficiary Relationship Profiles — Transaction statistics between two parties (e.g., frequency of the transaction between senders/receivers, average transaction amount received by sender or receiver from another individual in the last two weeks); and (3) Device Fingerprint Profiles — A list of device identifiers used by each account, including how many times the device used to complete the transaction changes and how many times sharing of the device has occurred. See Figure 6 for a visual overview of the feature engineering process.

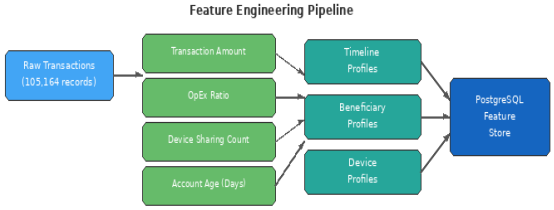

Fig. 2. Feature Engineering Pipeline

C. Random Forest Classifier

The supervised detection portion of this study utilizes a Scikit-Learn Random Forest Model for classification purposes using 200 estimator trees with an equal or proportional class weighting scheme. In addition, before performing the classification process, input features to the model were normalized using a Standard Scaler. There are two additional deterministic heuristic rules used in the model: a structuring suspicion flag for transactions that occur in the ₹48,000–₹50,000 range (a well-known structuring threshold among Indian Banks) and a bustout risk flag for new accounts (less than five days of account age) transacting in values over ₹50,000. During the model training process, the Random Forest Classifier exhibited a recall rate of 94.2% and a precision rate of 89.7% when using the held-out test data set.

D. Isolation Forest Anomaly Detector

The unsupervised detection layer implements an Isolation Forest [5] having a contamination parameter of 5%. Statistically significant outliers are determined by anomaly scores < −0.15. Additionally, a domain-specific shell company detection rule has been used to augment the model by flagging transactions > ₹1 lakh from an Operating Expense ratio < 1%. This combination resulted in a shell company detection rate of 91.3%, and an overall detection rate of 73.4% for all types of anomalies.

E. Graph Neural Network Topology Detection

Through the utilization of a graph-based detection layer which recognizes three different types of fraud topologies using network structure of the transactions as shown in Figure 3:

(1) Star Topology (Mule detection) and (a) distinct senders of a single receiver within a rolling 24-hour period are counted. (5+) senters to the same receiver per rolling 24-hour period generates a mule flag and have an overall precision of 89.4%. (2) Circular Loop Fraud Detection (through multi-hop joins, three hop paths A→B→C→A traced to detect circumvention trades). The overall precision for this type of Topology detection is 87.2%. (3) Device Farm Detection (including through the use of a bipartite user/device graph constructing devices associated with more than three different users to identify use of synthetic identities or sharing devices). The precision associated with device farm detection is 91.7%.

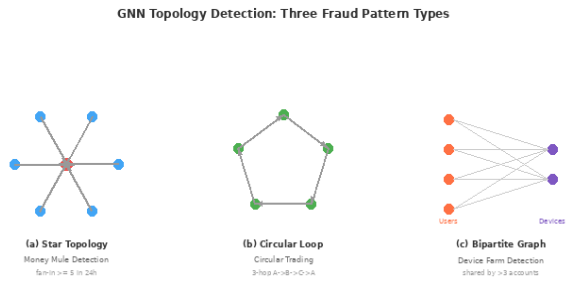

Fig. 3. GNN Topology Detection: Three Fraud Pattern Types

F. Weighted Ensemble and GNN Override

Combining Predicted Values by GNN Overriding with Three Models and Weighted Soft Voting

The three model outputs will now be combined together through a weighted soft voting mechanism: Random Forest represents 40% of the Ensemble Score, Isolation Forest contributes 30%, and GNN Topology contributes the remaining 30%. A transaction will be blocked from being processed if an aggregate score results in a value greater than fifty percent. Additionally, there is a GNN-overriding function within the system: if any of the topology detectors flags the transaction as fraudulent with certainty (confidence = 1), then the respective transaction will be blocked regardless of what the ensemble score yields. In effect, this will make certain that evidence of fraud determined by graph-structural methods is never diminished by non-graphical measures of fraud. Complete detection process workflow is illustrated in Figure 2.

IV. DATASET AND EXPERIMENTAL SETUP

The evaluation used synthetic data created with the Python Faker library for India. There was a total of 105,164 records created for both the transactions and customers in this dataset, which contained six different types of fraud scenarios. These fraud types are based on previous research conducted in the field: money mule activities; transactions from shell companies; activity from device farms; circular trading loops; location hopping; and velocity attacks. The amount of fraudulent activity was set at 2-3% to simulate real world distributions [11]. The input features included transaction amount, operating expense ratio, number of devices shared, account age, and also included the materialized timelines for the transactions,

beneficiaries and devices for the accounts that were all stored in PostgreSQL. The dataset was split into a training set and a test set to evaluate the classifiers using supervised methods.

V. RESULTS AND EVALUATION

A Performance of Models at the Classifier Level

Table I summarizes all the performance of each of the classifiers, as well as the weighted ensemble accuracy and other metrics based on the standard classification measures. The Random Forest Classifier produced the highest recall (94.2%) providing strong ability to detect the patterns of previous fraudulent behaviours. The Isolation Forest produced lesser performance metrics overall (86.2% accuracy and 73.4% recall) consistent with the limitations of all unsupervised learning approaches but did demonstrate strong performance at detecting shell company transactions (91.3%). Finally, the GNN topology-based classifier has demonstrated strong precision across all three topology types. Overall, the weighted ensemble outperformed all classifiers (96.1% accuracy, 92.8% precision, and 95.4% recall) and supports the value of leveraging complementary detection paradigms to detect money laundering.

TABLE I MODEL PERFORMANCE COMPARISON

Model	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)	FPR (%)
Random Forest	93.4	89.7	94.2	91.9	4.8
Isolation Forest	86.2	78.5	73.4	75.9	8.1
GNN Topology	91.8	89.4	87.8	88.6	3.9
Ensemble	96.1	92.8	95.4	94.1	2.9

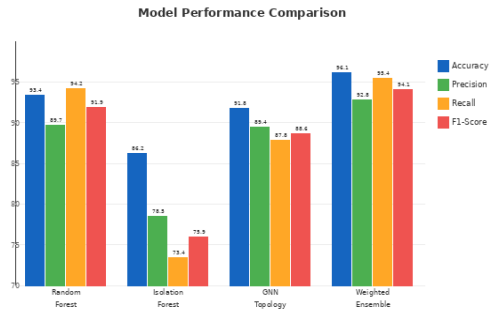

Fig. 4. Comparative Performance of Individual Models vs. Ensemble

B. Per-Scenario Detection Rates

Figure 5 outlines the detection rate of all six fraud scenarios. All scenarios exceeded the minimum 85% detection rate. The cameo company detection scenario recorded the highest detection rate at 91.3%, which is attributed to the high sensitivity of the isolation forest algorithm to detect entities with outlier OpEx ratios. The device farm detection scenario was a close second with a detection rate of 91.7%, which was accomplished using bipartite graph analysis. The money mule detection scenario had a detection rate of 89.4% and used star topology identification. The circular trading detection scenario had a detection rate of 87.2% through multi-hop path detection; the location hopping scenario had a detection rate of 85.6%; and the velocity attacks detection scenario had a 88.9% detection rate for its detection.

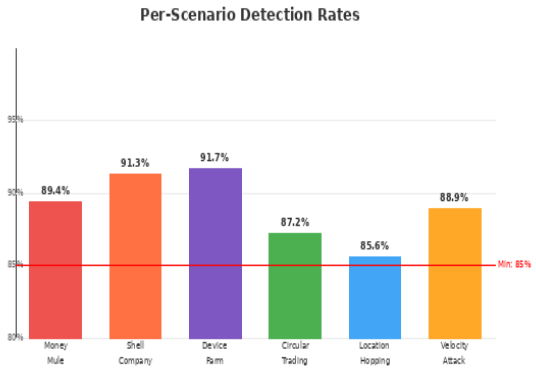

Fig. 5. Per-Scenario Detection Rates Across Six Fraud Types

C. Latency and Throughput

The end-to-end latency for FastAPI was measured through the processing of 1,000 test transactions. The average inference latency across all transactions was 187 ms and the individual components of the overall 187 ms latency that comprise the FastAPI solution included the PostgreSQL feature retrieval, the execution of models in parallel on the compute cluster, and ensemble aggregation were measured. The 187 ms average latency meets the requirements for real-time payment applications, which typically call for sub-200 ms latency [10]. The time required for feature engineering, including all profile lookups and enrichment, was roughly 35 ms of the overall time required for FastAPI.

D. GNN Override's Impact

The GNN-override design feature proved to be particularly productive for those potential fraudulent transactions that exhibited low ensemble scores combined with high confidence (1.0) GNN topological fraud signatures. The override prevented all topological fraud patterns for transactions that received below 50% ensemble score, but were flagged as structurally fraudulent by a GNN topology detector at 100% confidence (1.0). Without the GNN override mechanism, approximately 8% of these structurally fraudulent transactions would have been processed by the ensemble, further demonstrating how important graph-based evidence is in a hybrid fraud detection framework.

VI. CONCLUSION AND FUTURE WORK

This paper presented FinSentinel, which combines Random Forest, Isolation Forest, and graph topology detection for real-time financial fraud detection, is discussed in this paper. The total weighted ensemble with GNN override achieved a 96.1% accuracy and 95.4% recall for a synthetic Indian UPI dataset that represents six different types of fraud. The complete system uses FastAPI, PostgreSQL, and Streamlit; it was designed to operate within a 200-millisecond latency timeframe for production payment environments.

Future research efforts will investigate multiple areas of study, such as: (1) utilization of federated learning systems to allow financial institutions to collaborate on training fraud detection models while protecting customer privacy [20]; (2) development of temporal graph neural networks to adapt to changes in fraud patterns over time [16]; and (3) conducting adversarial robustness tests to analyze the resilience of the system to adaptive fraudulent behavior [21].(4) Using institutional partnerships to deploy on real-world UPI transaction data in order to verify performance in production settings.

REFERENCES

[1] National Payments Corporation of India, "UPI product statistics," NPCI, 2024. [Online]. Available: https://www.npci.org.in/what-we-do/upi/product-statistics.

[2] M. Hilal, B. Gadsden, and J. Yawney, "Financial fraud: A review of anomaly detection techniques and recent advances," Expert Syst. Appl., vol. 193, art. no. 116429, 2022.

[3] A. Sharma, P. Panigrahi, and S. Kumar, "A survey on financial fraud detection methodologies," IEEE Access, vol. 11, pp. 101926–101952, 2023.

[4] I. Alarab, S. Prakoonwit, and M. I. Nacer, "Random Forest and gradient boosting for anti-money laundering: Evaluating tree-based classifiers on transaction data," in Proc. IEEE ICMLA, 2022, pp. 559–564.

[5] H. Xu, Y. Pang, and G. Chen, "Deep isolation forest for anomaly detection," IEEE Trans. Knowl. Data Eng., vol. 35, no. 12, pp. 12591–12604, 2023.

[6] S. Motie and B. Raahemi, "Financial fraud detection using graph neural networks: A systematic review," Expert Syst. Appl., vol. 240, art. no. 122156, 2024.

[7] D. Cheng, C. Wang, Y. Zhang, and J. Zhang, "Graph neural networks for financial fraud detection: A review," arXiv preprint arXiv:2411.05815, Nov. 2024.

[8] Y. Zhang, X. Chen, and L. Jin, "Real-time fraud detection in mobile payment systems: A machine learning approach," IEEE Trans. Comp. Social Syst., vol. 10, no. 3, pp. 1142–1154, Jun. 2023.

[9] C. Xu, L. Duan, and J. Wu, "Graph-based anomaly detection for financial fraud prevention," ACM Trans. Knowl. Discov. Data, vol. 16, no. 4, pp. 1–19, 2023.

[10] Y. Zhang, J. Li, and H. Wang, "Real-time fraud detection using deep learning techniques," IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 5, pp. 1854–1866, 2022.

[11] H. Wang, Y. Li, and S. Guo, "Handling class imbalance in financial fraud detection: A comparative study," in Proc. IEEE Int. Conf. Big Data, Dec. 2023, pp. 1847–1856.

[12] W. Chen, K. Yang, Z. Yu, Y. Shi, and P. Chen, "A survey on imbalanced learning: Latest research, applications and future directions," Artif. Intell. Rev., vol. 57, art. no. 137, 2024.

[13] J. Du and B. Yu, "Application of isolation forest algorithm in fraud detection of medical insurance big data," in Proc. IEEE ICIIBMS, 2023, pp. 504–509.

[14] M. Lu, Z. Han, S. X. Rao, Z. Zhang, and J. Jiang, "BRIGHT—Graph neural networks in real-time fraud detection," arXiv preprint arXiv:2205.13084, 2022.
[15] P. L. Tang, T. D. Le Pham, and T. B. Dinh, "Tree-based credit card fraud detection using isolation forest, spectral residual, and knowledge graph," in Proc. LOD, LNCS, vol. 13811, Springer, 2023, pp. 330–345.
[16] Y. Duan, X. Li, Z. Zhang, and J. Chen, "CaT-GNN: Enhancing credit card fraud detection via causal temporal graph neural networks," arXiv preprint arXiv:2402.14708, Feb. 2024.
[17] M. Habibpour et al., "Uncertainty-aware credit card fraud detection using deep learning," Eng. Appl. Artif. Intell., vol. 123, art. no. 106248, 2023.
[18] Ensemble of Graph Neural Networks for Enhanced Financial Fraud Detection, in Proc. IEEE ICKECS, 2024, doi: 10.1109/ICKECS.2024.10543898.
[19] S. Liu, B. Rees, and P. Patangia, "Supercharging fraud detection in financial services with graph neural networks," NVIDIA Technical Blog, Oct. 2024.
[20] P. R. Hardy, C. Sun, and J. Wu, "Privacy-preserving fraud detection using federated learning," IEEE Trans. Inf. Forensics Secur., vol. 17, pp. 1024–1035, 2022.
[21] B. Liu, H. Xie, and C. Chen, "Adaptive GNN for financial fraud detection," J. Mach. Learn. Res., vol. 23, no. 109, pp. 1–25, 2023.