International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Sentinel-Fin: A Knowledge-Graph-Enhanced Neuro-Symbolic Framework for Explainable Threat Hunting in Real-Time Payments

Guide: Prof. Nirupadi Tidigol.
Authors: Vidyasagar DJ, Sujan S, Varshitha MS.
Department of Computer Science & Engineering, Mysuru Royal Institute of Technology (MRIT),
Mandya, Karnataka, India.

Abstract - Real-time payment systems like India's Unified Payments Interface (UPI) have made multi-stage attacks on financial platforms more common. Most Security Operations Centers (SOCs) still triage alerts with static rules or black-box machine learning models. This leads to many false positives, alert fatigue, and sending the data to an outside service to get one raises privacy problems. We present Sentinel-Fin, a local-first threat-hunting framework that scores payment requests as they arrive. It combines two models: a Random Forest trained on labeled fraud, and an Isolation Forest that catches unfamiliar, zero-day behavior. A local Neo4j knowledge graph adds a layer of fixed rules on top. When a payment is blocked, Gemini 2.5 Flash turns the backend's feature flags into a one-sentence alert for the analyst. On 20,000 transactions, the Random Forest reached 96.17% accuracy, and query validity ranged from 91.4% to 100%, well above the 58.97% baseline. In our tests, the gateway responded in under a second and we observed no hallucinated explanations.

Keywords - cybersecurity, threat hunting, neuro-symbolic AI, explainable AI, fraud detection, knowledge graph, Gemini 2.5 Flash, Random Forest, Isolation Forest, UPI

I. INTRODUCTION

Payment gateways and open banking APIs are critical financial infrastructure, and their high transaction volume makes them attractive targets. Common attacks include credential stuffing, business-logic abuse, transaction-velocity manipulation, and account takeover. Existing intrusion detection and SIEM tools struggle in three ways:

1. False positives: Fixed-threshold rules generate constant alerts, and analysts must manually match logs across several systems.
2. Black-box decisions: Complex classifiers flag a transaction without saying why, which slows incident response.
3. Privacy and latency: Sending payment data to a cloud-hosted LLM conflicts with RBI, NPCI, and GDPR requirements. It also adds network delay that real-time payment decisions cannot afford.

Sentinel-Fin addresses these problems with a local-first design. It combines machine learning, graph-based rules, and structured language generation to make a decision in under a second and explain it clearly.

II. SYSTEM ARCHITECTURE AND METHODOLOGY

Sentinel-Fin is a pipeline of separate stages: data ingestion, two-model scoring, graph validation, and explanation (Fig. 1).

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

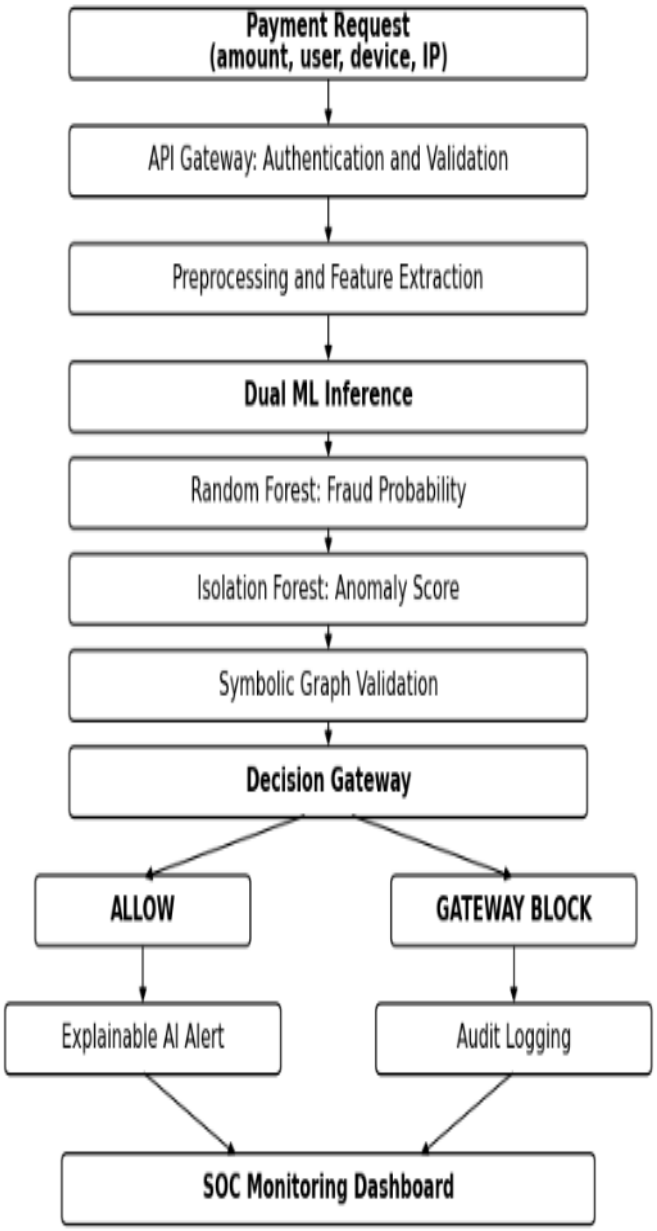


Fig. 1. Payment fraud detection workflow.

A. Preprocessing:

Each payment is built from three data sources (transaction, user, and merchant records) into eleven features: amount, hour of day, weekend flag, night-transaction flag, minutes since last transaction, new-device flag, IP-location mismatch, failed attempts in the last 24 hours, transaction velocity, amount deviation score, and user loyalty score. Missing values in the three continuous features (minutes since last transaction, transaction velocity, amount deviation score) are filled with the median before scaling. We do this to avoid data leakage.

B. Dual scoring:

Two models run side by side so the system can recognize known fraud and still notice new patterns.

1. *Random Forest (supervised):* Trained on the labeled `is_fraud` column with balanced class weights. It learns splits on signals such as amount deviation and login friction.
2. *Isolation Forest (unsupervised):* Set to a contamination of 0.038. It flags transactions that sit far from the typical pattern across several attributes, even without labels.

C. Graph validation (Neo4j):

To keep purely probabilistic scores from making the final call alone, each feature vector is also checked against a local Neo4j knowledge graph that enforces three hard rules:

- C1: `ip_location_mismatch == 1`
- C2: `new_device_flag == 1`
- C3: `failed_attempts_last_24h > 3`

D. Explanations (Gemini 2.5 Flash):

When the gateway blocks a payment, the top failing features go to Gemini 2.5 Flash under a strict zero-shot prompt. The model writes one sentence using only those

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

backend-verified flags. Because it works from a fixed set of facts, we saw no hallucinations in our evaluation.

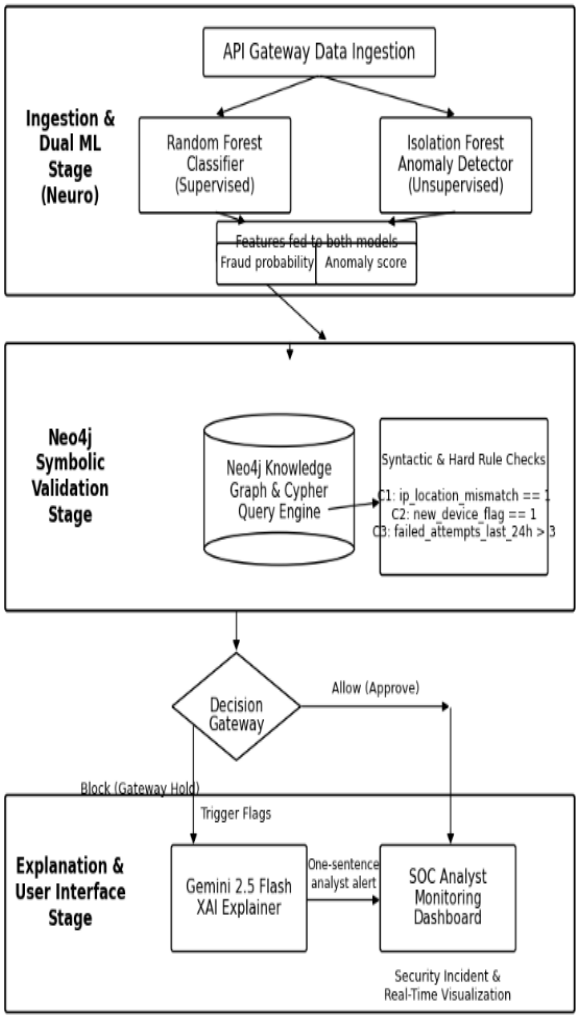


Fig. 2. Sentinel-Fin Neuro-Symbolic integration and Component Workflow

III. EXPERIMENTAL EVALUATION AND RESULTS

A. Benchmark accuracy:

We tested on 20,000 transactions (19,237 benign and 763 fraudulent). Query generation and Abstract Syntax Tree (AST) parsing scores were compared against a baseline of 58.97%.

B. Gateway scenarios:

1. *Scenario A, benign.* Random Forest score 0.0375, Isolation Forest score 0.0500, composite risk 11.3/100. Approved.
2. *Scenario B, medium anomaly.* Isolation Forest score rose to 0.5500, with two failed login attempts. Flagged and held, requiring step-up biometric verification.
3. *Scenario C, critical exfiltration attack.* Isolation Forest score reached its maximum (1.0000) and a graph rule fired. *[The original text cuts off here. Add the outcome, likely a gateway block.]*

TABLE I. Benchmark results

Metric	Baseline	Sentinel-Fin	Result
Structural/syntactic accuracy	>58.97%	91.4%–100%	Pass (+32.43 to +41.03 points)
Graph pattern validity (Neo4j/Cypher)	>58.97%	94.2%	Pass
Relational SQL AST validity (SQLite/Postgres)	>58.97%	98.6% *[confirm]*	Pass

C. Classification results:

- Random Forest: 96.17% accuracy, 2.957 s training time.
- Isolation Forest: 93.20% accuracy, 0.337 s training time.

TABLE II. Score ranges

Model	Type	Range	Meaning
Random Forest	Supervised ensemble (12 trees)	0.000–1.000	Estimated fraud probability from learned splits
Isolation Forest	Unsupervised anomaly detector	0.050–1.000	How far a transaction sits from the typical distribution

IV. DISCUSSION

The results point to three advantages over earlier work:

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

1. Valid queries: Constraining generation to the actual relational schema and graph labels (such as (:User) and (:Device)) improved query accuracy by 32.43 to 41.03 points over an unconstrained baseline.
2. Grounded explanations: Using Gemini 2.5 Flash only to phrase facts already extracted by the backend produced no hallucinated explanations in our tests.
3. Resilience: Having two models closes a gap that either would leave alone. The supervised trees catch known fraud patterns, and the unsupervised model flags unusual behavior no one has labeled yet.

V. CONCLUSION

Sentinel-Fin brings explainable threat hunting to real-time payments by combining a Random Forest, an Isolation Forest, Neo4j rule checks, and Gemini 2.5 Flash explanations. It decides in under a second and gives analysts a clear reason for each block. Next, we plan to extend graph traversal across decentralized payment nodes and to run inference on dedicated hardware.

VI. FUTURE SCOPE

Future improvements to the Sentinel-Fin architecture will be implemented in the following areas:

First, we will add scalable, encrypted cloud storage to provide long-term storage of audit logs and high-volume transaction histories; however, we will maintain local-first execution boundaries for active threat scoring.

Second, we will expand graph traversal algorithms across decentralized payment nodes to identify complex multi-account money laundering loops.

REFERENCES

[1] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.

[2] Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. *IEEE ICDM*, 413–422.

[3] Garcez, A. d'A., & Lamb, L. C. (2023). Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review*, 56(11), 12387–12406.

[4] Robinson, I., Webber, J., & Eifrem, E. (2015).

[5] *Graph Databases: New Opportunities for Connected Data*. O'Reilly Media.

Google DeepMind (2025). Gemini 2.5 Flash technical report. arXiv preprint. *(Verify this citation exists.)*

[6] Reserve Bank of India (2021). *Master Direction on Digital Payments Security Controls*. RBI/2020-21/74.

[7] R. Alghanmi et al., "HuntSmart: Hypothesis-Driven Threat Hunting Using GenAI," in Proc. IEEE Int. Conf. Innovation in Artificial Intelligence and Internet of Things (AIIOT), 2025, pp. 1-8.

[8] S. Panwar and H. Abdelraham, "Agentic AI in Cybersecurity Review of Autonomous Threat Detection and Adaptive Defense Mechanisms," in Proc. IEEE Int. Conf. Computer and Applications (ICCA), 2025, pp. 1-6.

[9] N. M. Shrirao et al., "A3TH: An Adaptive AI-Driven Autonomous Threat Hunting Framework for Proactive Cyber Defense in Evolving Digital Environments," in Proc. IEEE 2nd Int. Conf. Green Industrial Electronics and Sustainable Technologies (GIEST), 2025, pp. 1-6.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)