International Journal of Cyber Threat Intelligence and Secure Networking (IJCTISN)
eISSN: 3087-4289
Volume. 03, Issue. 05, pp. 18-26, May 2026"

Applying Graph Theory to Detect Collusive Fraud Rings in Distributed Reward Systems

Igor Litovsky
Founder & CTO, Mastermind Loyalty Toronto, Canada

Article Received: 15/03/2026, Article Accepted: 24/04/2026, Article Published: 06/05/2026
DOI: https://doi.org/10.55640/ijctisn-v03i05-01

ABSTRACT

Distributed reward systems circulate digital value through points, referral bonuses, coupon credits, vouchers, cashback balances, and partner-linked entitlements. Coordinated fraud in such systems often spreads across multiple accounts and shared infrastructure, weakening account-level screening. This article examines how graph theory can support the detection of collusive fraud rings in reward platforms with distributed value flows. The study draws on ten recent publications on graph anomaly detection, graph fraud detection, dynamic graphs, hypergraph learning, frequency-aware modeling, and interpretable graph analytics. The analytical section identifies the structural signatures of collusion, clarifies which graph representations expose them most effectively, and formulates a deployment model for ring-oriented detection. The article argues that collusive abuse becomes more legible when accounts, devices, addresses, payment instruments, referral links, and redemption paths are modeled as one evolving multi-entity graph. The proposed synthesis offers a practical basis for scoring, analyst review, and enforcement design in reward ecosystems.

KEYWORDS

graph theory, collusive fraud, fraud rings, distributed reward systems, graph anomaly detection, dynamic graphs.

Introduction

Distributed reward systems handle a form of digital value that moves across applications, merchants, payment channels, loyalty accounts, referral programs, and partner networks. Points, cashback units, promotional credits, and partner rewards retain measurable economic value and are susceptible to coordinated abuse. Controlled clusters of accounts can distribute activity across several identities, devices, payment instruments, and redemption targets. Each action may remain close to ordinary behavioral thresholds, while the group pattern reveals deliberate coordination.

Graph analysis provides a suitable framework for examining collusive fraud rings in distributed reward systems. The inquiry proceeds along three linked lines: the selection of graph representations that most clearly retain coordinated abuse in reward environments, the identification of graph-analytic procedures that distinguish collusion from intensive but legitimate user activity, and the design of an operational chain that connects ring detection with scoring, analyst review, and governance response. The article contributes a reward-specific analytical frame centered on collusive organization, shared infrastructure, and ring evolution across time windows. Its hypothesis states that collusive fraud in distributed reward systems leaves stable structural signatures in multi-entity graphs, and that these signatures become more visible when node-, edge-, community-, temporal-, and frequency-aware views are combined within a single detection framework.

Literature Review

The source base consists of ten peer-reviewed publications from 2021 to 2026 that address graph anomaly detection, graph neural fraud detection, multi-relation learning, dynamic collaborative fraud, promo-fraud adaptation, hypergraph modeling, frequency-aware graph analysis, and interpretable graph learning (Lou et al., 2025; Lu et al., 2026; Ma et al., 2023; Motie & Raahemi, 2024; Prasetya et al., 2026; Ren et al., 2023; Wang, Shen, & Dong, 2025; Wang, Liu, Liu, & Liu, 2023; Zhao & Chen, 2026; Zhang et al., 2021). Screening followed three criteria. Each source had to address fraud, anomaly, or coordinated abuse through graph-structured reasoning. Each source had to contribute one of the building blocks required for the present topic, namely anomaly typology, multi-entity graph construction, relation-sensitive learning, temporal evolution, high-order interaction modeling, class imbalance, camouflage resistance, or interpretability. Each source also had to remain usable for an analytical article without experimental replication. The corpus covers five thematic groups: graph anomaly typologies and anomaly levels; graph-based fraud taxonomies in financial settings; camouflage and relation-sensitive node learning; dynamic or adversarial graph evolution; and interpretable or higher-order graph modeling.

Methodology

The methodological design is based on a structured review of recent research on graph anomaly detection, fraud-oriented graph learning, dynamic graph modeling, and higher-order relational representations. The reviewed studies are compared by graph type, fraud unit of analysis, treatment of time, handling of relation diversity, resistance to camouflage, and degree of interpretability. Their findings are then reorganized from the perspective of distributed reward systems, in which suspicion accumulates at the level of a coordinated ring. On that basis, the article differentiates ring archetypes, matches them to the graph view that most clearly exposes them, and derives an operational sequence from graph assembly to analyst-readable case output.

Results

Recent work on graph-oriented fraud detection converges on a point with direct relevance for reward systems. Collusive abuse becomes harder to detect when fraudulent actors split their activity across several entities and reuse shared infrastructures in controlled doses. In such cases, account-level screening captures fragments of the pattern, while the graph preserves the full structure of coordination (Ma et al., 2023; Motie & Raahemi, 2024; Zhang et al., 2021). Reward fraud cannot be reduced to a transaction, a login, or a single account. The more informative unit is the relation set formed by accounts, devices, payment instruments, referral links, shipping destinations, IP addresses, merchant touchpoints, and redemption targets.

A survey by Ma et al. (2023) distinguishes node anomalies, edge anomalies, subgraph anomalies, and graph-level anomalies. Collusive reward abuse aligns most closely with the subgraph level, because a fraud ring usually appears as a connected structure of mildly suspicious entities joined by repeated reuse of shared infrastructure. Motie and Raahemi (2024) reach a similar conclusion based on the fraud-detection literature. Their review notes that many graph fraud studies still concentrate on supervised node classification, while edge-level and graph-level anomaly detection receive less attention. Distributed reward systems expose the cost of that imbalance. A referral account with normal redemption intervals or a coupon account with moderate earning activity may still belong to a suspicious ring when the same sponsor, address, device, or payout destination recurs across a local cluster.

This line of reasoning leads to a representation requirement. A reward platform needs a heterogeneous graph with typed nodes and typed edges. In operational terms, such a graph is built from concrete entity types. Nodes can represent loyalty accounts, devices, payment tokens, referral codes, shipping addresses, merchants, coupon families, and payout endpoints, while edges record shared logins, reuse of the same payment instrument, sponsor–invitee links, coupon redemption paths, or settlement to the same withdrawal channel. A referral farming ring, for instance, often stays below account-level thresholds yet becomes visible once many invitee accounts are activated from a narrow device pool and redeem through one payout path. A cashback extraction ring follows a different geometry: value accumulates across several accounts and converges on a small set of merchant or withdrawal endpoints. Accounts, devices, payment tokens, referral codes, payout endpoints, merchants, addresses, and promotional artifacts carry different evidentiary weight. Research on relation-sensitive fraud detection supports this requirement. Wang, Liu, Liu, and Liu (2023) show that multi-relation learning strengthens node representations when suspicious behavior is distributed unevenly across different link types. An account-to-device edge has a different investigative value from an

account-to-referral edge or an account-to-redemption-target edge. Collusive actors often exploit that difference. They spread actions across several relation types so that no single channel appears extreme, while the combination of channels reveals a coherent fraud pattern.

Camouflage enters the graph at this point. Zhang et al. (2021) describe fraud detection under graph inconsistency and imbalance, with camouflage defined as fraudulent nodes' deliberate efforts to blend into benign neighborhoods. Lou et al. (2025) sharpen that idea by distinguishing context-related behavior from camouflage-related behavior. Their work shows that time intervals, out-degree structure, and adaptive aggregation help expose disguised fraud nodes in financial graphs. Reward systems exhibit analogous behavior. A collusive referral ring may mix low-value legitimate actions with synchronized reward extraction. A coupon ring may maintain ordinary browsing and occasional purchases while rotating accounts through the same device pool and redemption path. Graph aggregation, therefore, needs to preserve suspicious local structure without allowing benign neighbors to dilute it.

The literature becomes most informative when several sources are read together on the same question. Ma et al. (2023) classify anomalies by structural level and temporal setting. Motie and Raahemi (2024) show that the financial fraud literature still favors node-level detection over edge-level or graph-level reasoning. Wang, Shen, and Dong (2025) move beyond pairwise structure and argue that multi-hop collaborative attacks require hypergraph representations to capture higher-order group behavior. Prasetya et al. (2026) introduce adversarial evolution by showing that graph fraud patterns shift across rounds and that models trained on one graph state become less reliable as fraudsters adapt. Read together, these studies support a coherent rule for reward systems: collusive ring detection should begin with multi-entity graph construction, proceed to subgraph and community analysis, incorporate high-order representations when a single fraud episode binds multiple entities, and preserve temporal memory across campaign rounds.

High-order structure deserves separate attention because many reward fraud episodes involve more than pairwise coordination. A referral ring may contain one sponsor account, many controlled invitees, one shared device pool, a small number of payout endpoints, and a sequence of redemptions routed through the same merchant set. A pairwise graph approximates this configuration through many edges. A hypergraph encodes the co-participation pattern more directly by binding multiple entities into a single interaction unit. Wang, Shen, and Dong (2025) argue that multi-hop collaborative attacks remain difficult to expose when models assume only ordinary pairwise structure. Reward systems provide a natural setting for that claim. Promotional abuse often emerges through one joint episode that links accounts, devices, coupon families, and redemption destinations. Hypergraph modeling preserves that episode structure with higher fidelity than a flat pairwise projection.

Temporal structure adds another layer of evidence. Lou et al. (2025) show that time intervals and out-degree information contribute useful context for fraud detection. Ren et al. (2023) examine collaborative fraudsters in dynamic graphs and find that graph evolution carries information that static graph models fail to encode. Prasetya et al. (2026) reach a similar conclusion in an adversarial multi-round setting, where the graph evolves as fraudsters respond to detection pressure. Zhao and Chen (2026) extend the discussion into the frequency domain and show that graph fraud detection benefits from learning across both homophilic and heterophilic relations. For reward systems, these findings separate two temporal forms of collusion. One form appears as long-lived low-frequency coordination marked by stable communities, repeated infrastructure reuse, and persistent ring cohesion. The second form appears as high-frequency bursts marked by referral cascades, synchronized coupon redemption, and dense exploitation within short campaign windows. A reward detector that captures only persistent communities will miss short campaign swarms. A detector that focuses only on bursts will miss durable account mule networks.

Interpretability remains central because reward operators need case narratives. Lu et al. (2026) address class imbalance and interpretability by leveraging a graph learning framework that makes spectral responses and node-level evidence more transparent. Zhao and Chen (2026) also contribute to this question by showing how different graph frequencies affect fraud sensitivity. These studies matter in reward environments where analysts must explain why a referral tree was frozen, why a payout cluster was delayed, or why a redemption chain was escalated. Structural evidence for such decisions can be built from repeated shared entities, short cycles, unusually dense local communities, stable bridge nodes, and the recurrence of the same motifs across adjacent

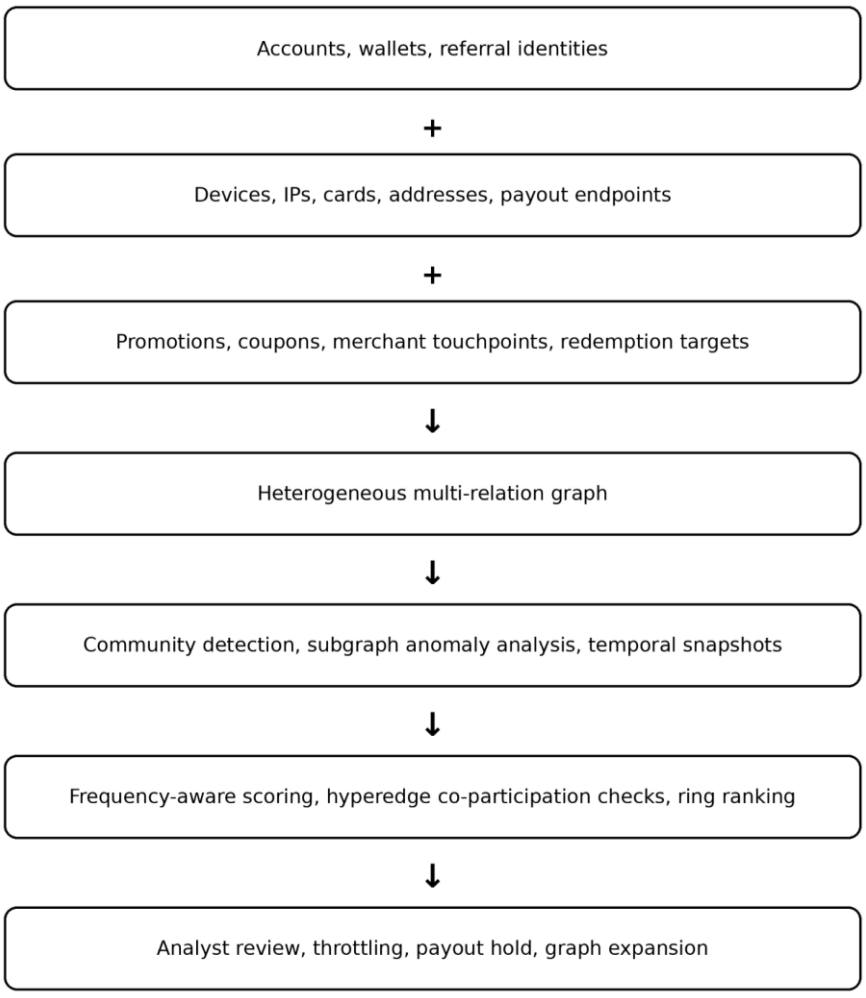
time windows. Spectral interpretation adds another dimension by clarifying whether the model responds to sharp local irregularities or to broader cluster organization.

Class imbalance complicates ring detection in a way that aligns closely with reward systems. Fraud rings tend to be small, sparse, and only partly labeled. Reward abuse intensifies the problem because one ring can generate many low-loss actions before any single action reaches an investigator or receives confirmed labeling. Zhang et al. (2021), Wang, Liu, Liu, and Liu (2023), and Lu et al. (2026) each address imbalance in graph fraud settings through different mechanisms, including inconsistency-aware modeling, relation-sensitive learning, adaptive sampling, and contrastive structure learning. Their combined implication is methodological. Reward platforms gain more stable results when they combine graph construction, weakly supervised community ranking, local anomaly scoring, and then targeted classification of nodes or clusters. Fully supervised ring detection remains useful, but it relies too heavily on timely labels to serve as the sole detection path.

The analytical picture that emerges from these studies is broader than a comparison of algorithms. Reward fraud becomes structurally legible when a platform builds a heterogeneous graph, examines dense local coordination and bridge behavior, tracks recurrence across time windows, and evaluates whether several entities co-participate in the same fraud episode. The suspicious unit then shifts from the single account to the coordinated structure. That shift improves both detection logic and investigative logic. Analysts evaluate whether the account falls within a ring defined by repeated entity reuse, motif recurrence, temporal synchrony, and persistence across adjacent campaign rounds.

Figure 1 summarizes that logic in a reward-oriented sequence.

Figure 1. Graph-theoretic detection logic for collusive fraud rings in distributed reward systems. Adapted from Prasetya et al. (2026)



The figure illustrates one operational implication of the literature. Graph construction, structural scoring, and response logic need to remain connected within a single detection pipeline. Prasetya et al. (2026) and Ren et al. (2023) show that fraud graphs change under pressure, indicating that the post-intervention topology becomes part of the next detection cycle. A ring detector for reward systems benefits from iterative use. Referral freezes, payout holds, or campaign throttles do not end the analytical process. They reveal whether suspicious clusters dissolve, migrate, fragment, or reconnect through new bridges. Those topological changes provide analysts with additional evidence of ring cohesion and adaptation.

The sources reviewed here also support a practical deployment logic for distributed reward systems. The first layer defines graph entities, relation types, and time windows. The second layer examines local communities, suspicious motifs, bridge nodes, and shared infrastructure. The third layer tracks whether a cluster persists or reorganizes after a control change. The fourth layer turns graph output into analyst-readable case narratives. Lu et al. (2026) make that final layer especially relevant because graph-based fraud scoring has limited operational value when the investigator cannot identify the structural reason for escalation. Reward environments need accurate ring detection and evidence that an analyst can audit.

Discussion

Graph-based ring detection fits distributed reward systems because collusive abuse is organized through relations, timing, and repeated entity reuse. Rule engines still matter in this setting. They catch immediate policy violations, block obvious coupon abuse, and interrupt extreme velocity spikes. They evaluate a redemption, a login, or a referral action. A ring detector evaluates how those events connect across the program's wider structure. That change strengthens investigative accuracy and makes enforcement more consistent.

A workable implementation model begins with entity normalization. At the implementation layer, the workflow can be specified more concretely. NetworkX is suitable for early graph inspection, rule debugging, and verifying connected components, short cycles, bridge nodes, and local community density. Once the graph moves from analytical prototyping to production-scale processing, PyTorch Geometric or DGL provide a more practical environment for heterogeneous and multi-relational learning over accounts, devices, addresses, payment instruments, referral links, and payout destinations. Within the GNN family, GCN offers a stable baseline for detecting densely connected reuse patterns, whereas GAT is more effective when different edge types carry unequal evidentiary weight and the model must attend selectively to account–device, account–address, account–merchant, or account–payout relations. Under this setup, graph-based ring detection becomes a technically executable pipeline that links graph construction, structural scoring, cluster ranking, and analyst review.

Teams need stable identifiers for devices, payout channels, referral codes, shipping destinations, payment instruments, merchant touchpoints, and promotional artifacts. The second stage builds rolling graph windows with a defined freshness policy for each relation. The third stage scores clusters through a ring suspicion logic that combines shared-entity concentration, motif recurrence, bridge centrality, and temporal compression. The fourth stage assigns a response. High-confidence rings move to payout holds, referral freezes, or campaign-level throttling. Medium-confidence rings are moved to graph expansion analysis so that adjacent communities and second-order neighbors can be checked before more drastic action is taken. Low-confidence cases stay under observation and continue to accumulate evidence.

Table 1 compares the main ring archetypes likely to appear in distributed reward systems and identifies the graph view that best exposes each pattern.

Table 1. Ring archetypes in distributed reward systems and the graph view that best exposes them

Ring archetype	Typical coordination pattern	Dominant graph view	Strongest structural clue	Primary operational response
Referral farming ring	One sponsor or a small sponsor cluster linked to many controlled invitees	Directed multi-relation graph	Shallow trees with repeated shared entities and compressed invite-to-redeem timing	Freeze referral payouts and expand the one-hop graph
Cashback extraction ring	Multiple accounts route earning activity into a narrow redemption or withdrawal channel	Temporal heterogeneous graph	Convergent paths toward the same payout endpoint or merchant pair	Delay settlement and isolate the payout cluster
Coupon stacking ring	Accounts rotate coupon use across devices and addresses	Bipartite or tripartite graph	Recurrent reuse of promotion artifacts across nominally separate identities	Suspend the coupon family and audit linked nodes
Account mule ring	Accounts receive value from several sources and channel it outward	Community graph with bridge analysis	Dense internal reuse and a small number of repeated outward bridges	Hold transfers and inspect bridge nodes first
Promo abuse swarm	Newly created identities exploit one campaign in a short burst	Dynamic graph with burst analysis	Sudden local density growth within one campaign window	Rate limit campaign actions and preserve graph snapshots
Hybrid collusive ring	Referral, coupon, and redemption abuse appear within one structure	Hypergraph or layered graph	The same actors co-participate across several fraud episodes	Escalate to a full ring-level case review

The comparison suggests that ring detection performs best when the graph view aligns with the geometry of the abuse pattern. Referral fraud depends on the direction of branching and rapid sponsor-invitee turnover. Cashback extraction concentrates value flows into a narrow set of endpoints. Coupon abuse is driven by repeated reuse of artifacts. Mule behavior becomes legible through bridge concentration between suspicious communities and outward destinations. Hybrid rings require layered or high-order representations because one group recurs across multiple fraud episodes. A single static graph remains workable for coarse surveillance, whereas a modular graph stack provides analysts greater precision.

Decision logic needs the same discipline. A platform gains little from moving directly from graph suspicion to hard enforcement. Shared households, corporate networks, public connectivity, and intense but legitimate campaign participation can still produce ambiguous structures. A staged threshold model works better. The first threshold places a cluster under structured observation. The second threshold introduces friction, such as redemption delay, referral caps, or temporary coupon throttling. The third threshold opens a case, freezes value movement, or suspends the relevant campaign path. This sequence preserves investigative control and reduces avoidable false positives.

Monitoring becomes indispensable once such a detector enters production. Teams need a small set of metrics that measure structural quality, operational yield, and review usability (see Table 2).

Table 2. Monitoring metrics for graph-based ring detection in distributed reward systems

Monitoring domain	Metric	What the metric captures	Warning sign	Follow-up action
Structural quality	Shared-entity concentration per flagged cluster	Whether the case rests on real infrastructure reuse or weak coincidence	Concentration declines across recent flags	Review entity normalization and edge rules
Community behavior	Cluster persistence across adjacent windows	Whether the ring remains coherent after friction	Clusters disappear after minimal policy change	Test for campaign noise and revise time windows
Operational value	Loss prevented per escalated ring	Whether ring-level review produces a material benefit	Review volume rises while the prevented loss stays low	Raise escalation thresholds or narrow graph expansion
Precision under friction	Share of delayed actions later cleared as legitimate	Whether controls generate an unnecessary burden	Clearance rate stays high over several review cycles	Reduce automated throttling and tighten analyst triage
Model stability	Drift in ring-score distribution across campaigns	Whether the detector remains calibrated under changing program logic	Scores inflate during campaigns without confirmed abuse	Recalibrate scoring and add campaign-aware normalization
Explainability	Share of cases with analyst-readable structural evidence	Whether investigators can justify escalation	Many flagged cases lack a clear topology narrative	Enrich case summaries with motifs, bridges, and shared-entity evidence

These metrics pull evaluation away from abstract benchmark language and back toward operational consequences. Shared-entity concentration tests whether the detector flags real reuse patterns. Cluster persistence tests whether graph friction reveals durable coordination or incidental campaign density. Prevented loss per escalated ring measures economic value. Clearance rate measures customer and analyst burden. Explainability coverage measures whether the system can survive institutional review. A detector with strong latent performance and weak case narratives will lose analyst trust over time.

Graph-theoretic ring detection also has value before model output reaches the fraud queue. It can guide policy design. If most damaging clusters concentrate around referral codes and shared payout channels, teams can add friction to those paths. If campaign abuse occurs within the first hours of a launch, teams can set stage limits and

reduce promotional throughput during the opening phase. If the same device pool reappears across nominally unrelated accounts, identity trust policies can shift from account-level checks toward infrastructure-level checks. A ring detector becomes more useful when it informs control placement.

A reward operator benefits from a sequence that starts before model training. Teams first define which entities carry the highest value risk. They then choose the graph layers that preserve those entities and their relations. Only after that stage do they select methods for community discovery, subgraph anomaly scoring, hypergraph analysis, or frequency-sensitive learning. Graph methods inherit the assumptions built into graph construction. Weak entity design narrows the evidentiary value of even strong models. Careful entity design gives moderate models room to perform well in production.

A mature reward program can turn graph-based ring detection into a stable operational discipline. Graph construction stores structural memory. Community analysis identifies a coordinated organization. Temporal tracking reveals persistence, migration, or fragmentation. Case synthesis translates graph evidence into governance action. That sequence fits the nature of collusive abuse, which unfolds across relations and time.

Conclusion

Collusive fraud in distributed reward systems becomes more legible when the platform models accounts, devices, referral paths, payment instruments, destinations, and promotional artifacts as one evolving multi-entity graph. Graph structures that preserve relation type, community topology, and temporal evolution give investigators a stronger basis for recognizing coordinated abuse than flat account-level records.

The reviewed studies indicate that collusion leaves a set of recurring structural signals. Subgraph anomalies, repeated shared-entity reuse, bridge concentration, temporal compression, hyperedge co-participation, and frequency-sensitive graph behavior expose ring organization with greater precision than isolated node scoring. Camouflage and class imbalance remain serious constraints, yet the literature provides workable paths through relation-sensitive learning, adaptive sampling, dynamic graph analysis, and interpretable graph modeling.

A deployable ring-detection framework for reward systems needs staged graph assembly, ring-level ranking, calibrated response thresholds, readable case narratives, and monitoring metrics tied to structural quality and operational value. The hypothesis stated in the introduction is supported. Collusive fraud in distributed reward systems leaves persistent graph signatures, which become more visible when node, edge, community, temporal, and frequency-aware perspectives are combined within a single analytical framework.

References

1. Lou, C., Wang, Y., Li, J., Qian, Y., & Li, X. (2025). Graph neural network for fraud detection via context encoding and adaptive aggregation. Expert Systems with Applications, 261, 125473. https://doi.org/10.1016/j.eswa.2024.125473

2. Lu, J., Xu, Q., & Hu, J. (2026). A novel graph learning framework for interpretable and imbalance financial fraud detection. Engineering Applications of Artificial Intelligence, 167, 113709. https://doi.org/10.1016/j.engappai.2025.113709

3. Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q. Z., Xiong, H., & Akoglu, L. (2023). A comprehensive survey on graph anomaly detection with deep learning. IEEE Transactions on Knowledge and Data Engineering, 35(12), 12012–12038. https://doi.org/10.1109/TKDE.2021.3118815

4. Motie, S., & Raahemi, B. (2024). Financial fraud detection using graph neural networks: A systematic review. Expert Systems with Applications, 240, 122156. https://doi.org/10.1016/j.eswa.2023.122156

5. Prasetya, H. A., Liu, X., Murata, T., & Matono, A. (2026). A multi-rounded adversarial scenario for graph-based promo fraud detection. Social Network Analysis and Mining, 16(1), 24. https://doi.org/10.1007/s13278-025-01566-0

6. Ren, L., Hu, R., Li, D., Liu, Y., Wu, J., Zang, Y., & Hu, W. (2023). Dynamic graph neural network-based fraud detectors against collaborative fraudsters. Knowledge-Based Systems, 278, 110888. https://doi.org/10.1016/j.knosys.2023.110888

7. Wang, Q., Shen, Y., & Dong, H. (2025). Hypergraph-based contrastive learning for enhanced fraud detection. Frontiers in Artificial Intelligence, 8, 1703135. https://doi.org/10.3389/frai.2025.1703135

8. Wang, X., Liu, Z., Liu, J., & Liu, J. (2023). Fraud detection on multi-relation graphs via imbalanced and interactive learning. Information Sciences, 642, 119153. https://doi.org/10.1016/j.ins.2023.119153

9. Zhang, G., Wu, J., Yang, J., Beheshti, A., Xue, S., Zhou, C., & Sheng, Q. Z. (2021). Fraudre: Fraud detection dual-resistant to graph inconsistency and imbalance. 2021 IEEE International Conference on Data Mining (ICDM), 867–876. https://doi.org/10.1109/ICDM51629.2021.00098

10. Zhao, W., & Chen, H. (2026). Learning frequency-aware graph fraud detection. Neural Networks, 198, 108600. https://doi.org/10.1016/j.neunet.2026.108600