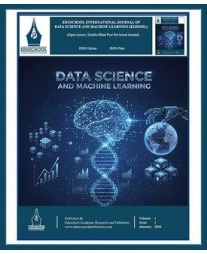
EDUSCHOOL INTERNATIONAL JOURNAL OF DATA SCIENCE AND MACHINE LEARNING (EIJDSML)

(Open Access, Double-Blind Peer Reviewed Journal)

ISSN Online:



Graph Neural Networks for Financial Fraud and Anomaly Detection

Raji N

Assistant Professor, Department of Computer Science, YuvakshatraInstitute of Management Studies (YIMS), Mundur,India

Article information

Received: 13th April 2026
Received in revised form: 14th May 2026
Accepted: 15th June 2026
Available online: 18th July 2026

Volume: 1
Issue: 2
DOI: https://doi.org/10.5281/zenodo.21412086

Abstract

Financial fraud now spreads through coordinated accounts whose risk is visible mainly in how they connect rather than in any single record. Tree based classifiers that score transactions in isolation miss this relational signal. This paper reviews and consolidates how graph neural networks (GNNs) cast fraud and anomaly detection as node and edge classification over transaction graphs, where message passing propagates evidence among accounts, devices, and merchants. We describe the main architectures used in practice, namely graph convolutional networks, GraphSAGE, and graph attention networks, and explain how each aggregates neighbourhood information. Two problems dominate the fraud setting: extreme class imbalance, since genuine fraud is rare, and camouflage, where fraudsters attach to honest neighbours to dilute their signature. We summarise label aware and neighbour selecting designs such as CARE-GNN and PC-GNN that address both. Heterogeneous and temporal graph variants are discussed for settings with several node types and evolving relations, alongside sampling methods that keep training tractable on large graphs. Using illustrative experiments patterned on public benchmarks (Elliptic, Yelp and Amazon review fraud, and IEEE-CIS), we report AUC, AUPRC, F1, and recall at k, and show that imbalance aware GNNs improve minority recall over a gradient boosted baseline. We close with notes on explainability and the engineering needed for production scoring.

Keywords:- Graph Neural Networks, Fraud Detection, Anomaly Detection, Message Passing, Class Imbalance, Heterogeneous Graphs, GraphSAGE, Graph Attention Network.

.

I. INTRODUCTION

Payment networks, online marketplaces, and lending platforms process billions of events each day, and a small fraction of those events are fraudulent. Card testing, account takeover, money laundering, and fake reviews share a common trait: the offending accounts rarely act alone. They reuse devices, shared funding instruments, and mule accounts, forming dense subgraphs that look ordinary one record at a time. Classical detectors built on gradient boosted trees treat each transaction as an independent feature vector and therefore overlook these relations [1], [2].

Representing the data as a graph changes the question. Accounts, cards, devices, and merchants become nodes; payments, logins, and shared attributes become edges. Fraud detection then turns into node or edge classification, and anomaly detection into the search for nodes whose neighbourhood differs from the norm. Graph neural networks (GNNs) learn node representations by passing messages along edges, so a node embedding

reflects both its own attributes and the company it keeps [3], [4], [5]. This relational view has produced strong results on public fraud benchmarks and is now common in industrial risk systems [6], [7].

Two properties of fraud data make the modelling harder than standard node classification. First, labelled fraud is rare, often well under two percent of nodes, so a model can reach high accuracy while missing most fraud. Second, fraudsters camouflage by transacting with many honest accounts, which weakens the very neighbourhood signal a GNN relies on [8], [9]. Methods such as CARE-GNN and PC-GNN respond by selecting informative neighbours and rebalancing the sampled graph. This paper organises these ideas and reports illustrative metrics that mirror published benchmark behaviour; the numbers are for exposition and do not describe a deployed system.

The remainder of the paper is structured as follows. Section II covers graph formulations of fraud and related work. Section III presents the core GNN architectures. Section IV addresses imbalance and camouflage. Section V describes datasets, methodology, and results. Section VI discusses explainability and deployment, and Section VII concludes.

II. BACKGROUND: GRAPHS FOR FRAUD AND RELATED WORK

A. Transaction Graphs

A transaction graph G = (V, E) holds entities in V and their interactions in E. In a payment setting a node may be a customer, a card, a device fingerprint, or a merchant, while an edge records a payment or a shared attribute. Each node carries a feature vector (transaction counts, amounts, velocity, account age), and each node may carry a label: fraud or legitimate. Fig. 1 sketches such a graph, with a dense fraud ring on the right and camouflage links that connect fraudulent accounts to honest neighbours.

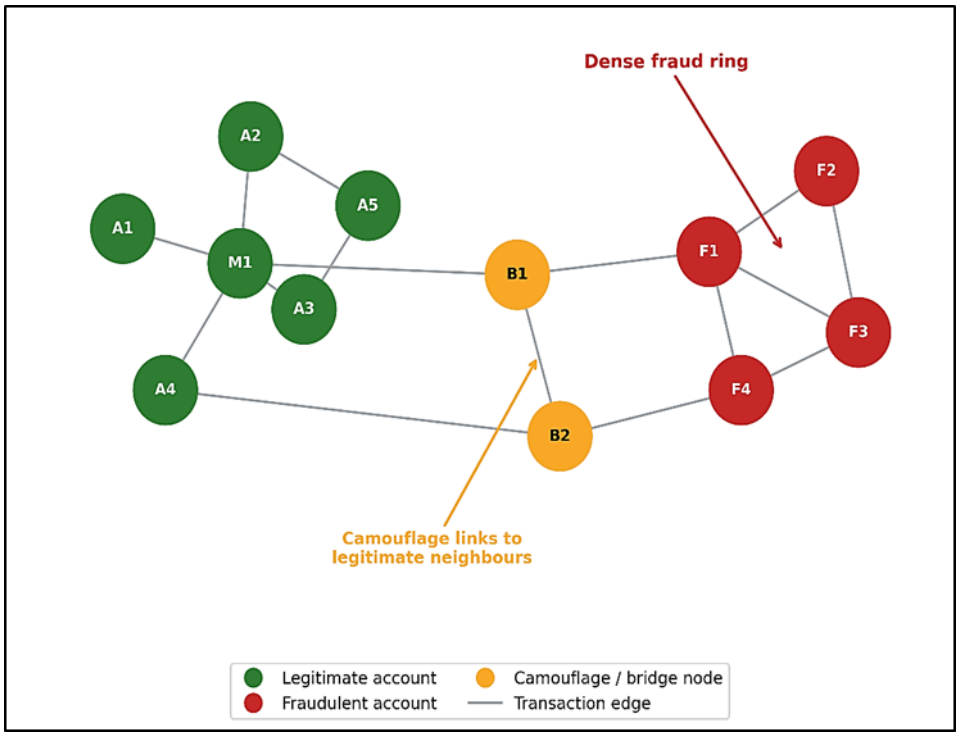


Fig. 1: Transaction graph with legitimate (green), fraudulent (red), and camouflage (amber) accounts. Fraud forms a dense ring, while camouflage edges connect it to honest neighbours.

Fraud signals appear at several scales. At the node level, an account may show unusual velocity. At the edge level, a single transfer may link two otherwise unrelated communities. At the subgraph level, a ring of accounts may cycle funds among themselves. A useful detector must read all three, which is why representation learning on graphs is attractive.

B. Related Work

Early graph based fraud detection used hand built features such as degree, triangle counts, and PageRank, fed to a downstream classifier [10]. Spectral and dense subgraph methods detected suspicious communities without labels [11]. The shift to end to end GNNs followed the graph convolutional network of Kipf and Welling [3] and the inductive GraphSAGE model of Hamilton et al. [4], which made training on large, growing graphs practical. Velickovic et al. introduced attention over neighbours with the graph attention network [5]. For fraud

specifically, Dou et al. proposed CARE-GNN to counter camouflage [8], and Liu et al. proposed PC-GNN to address imbalance through choice aware sampling [9]. Surveys by Pourhabibi et al. [12] and Motie and Raahemi [13] catalogue the wider space, and Weber et al. released the Elliptic Bitcoin dataset that anchors much of the empirical work [6].

III. GNN ARCHITECTURES FOR FRAUD

A GNN layer updates each node by collecting messages from its neighbours, combining them, and transforming the result. Stacking L layers lets information travel up to L hops, so a node sees an expanding neighbourhood. Fig. 2 shows the pipeline: sample neighbours, form messages from node and edge features, aggregate, update through a small network, and repeat per layer before a softmax produces a fraud score.

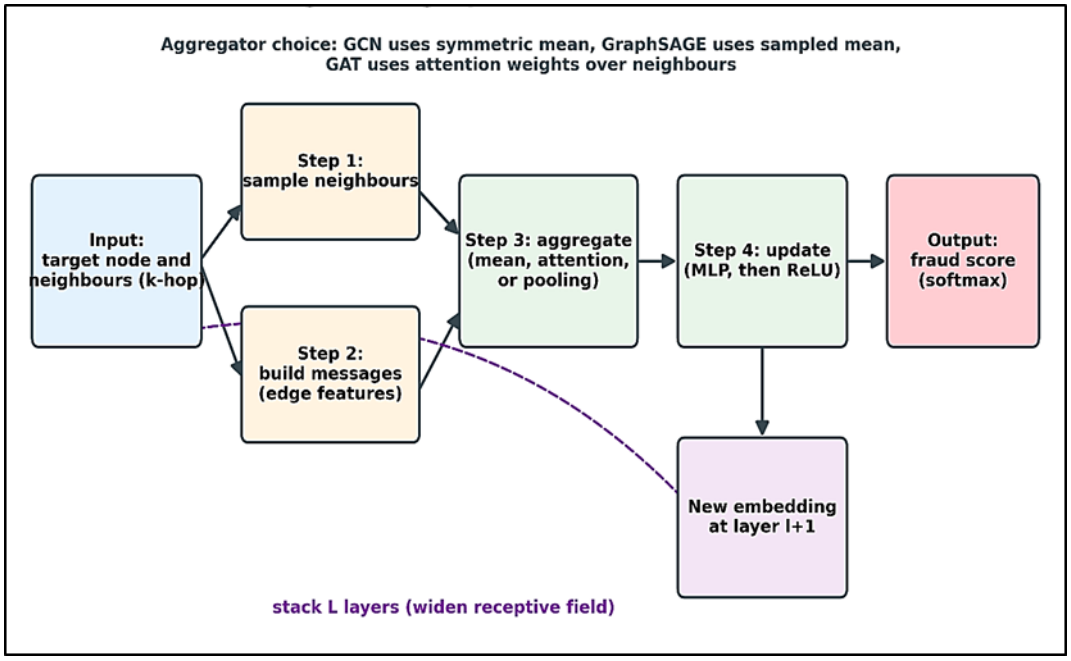


Fig. 2: Message-passing pipeline. Neighbours are sampled, messages are aggregated by mean, attention, or pooling, and an update network produces the next embedding; L layers widen the receptive field.

A. Graph Convolutional Network

The graph convolutional network (GCN) averages neighbour features with a symmetric normalisation that down weights high degree nodes [3]. It is simple and fast but treats every neighbour as equally trustworthy, which is a weakness when fraudsters attach to honest accounts. GCN also assumes the full graph is available at training time, which limits its use on streaming data.

B. GraphSAGE

GraphSAGE learns aggregation functions and samples a fixed number of neighbours per node, so it scales to large graphs and generalises to nodes unseen during training [4]. Mean, pooling, and LSTM aggregators are common. The inductive setting matches fraud systems, where new accounts appear constantly and must be scored without retraining the whole model.

C. Graph Attention Network

The graph attention network (GAT) assigns a learned weight to each neighbour, so the model can emphasise the few relations that carry risk and discount the rest [5]. This selectivity helps against camouflage, since honest neighbours can receive low attention. Multi head attention stabilises training. The cost is extra computation per edge, which sampling can offset.

IV. HANDLING IMBALANCE AND CAMOUFLAGE

Two obstacles separate fraud graphs from ordinary node classification. Imbalance means the positive class is a tiny minority, so a model that predicts legitimate everywhere scores well on accuracy yet finds no fraud. Camouflage means fraudsters connect to honest neighbours and copy honest features, blurring the signal that message passing would otherwise sharpen [8], [9].

A. Cost Sensitive and Resampling Strategies

Standard remedies reweight the loss toward the minority class, or resample the graph so each batch contains enough fraud. Focal loss concentrates gradient on hard examples [14]. Graph aware oversampling such as GraphSMOTE synthesises minority nodes within the embedding space rather than the raw feature space [15]. These help, but naive oversampling can also amplify camouflage noise if it copies the diluted neighbourhoods.

B. Neighbour Selection: CARE-GNN

CARE-GNN treats aggregation as a decision. A label aware similarity measure ranks neighbours, a reinforcement learning module selects how many to keep, and aggregation runs only over the chosen set [8]. By filtering out camouflage neighbours before averaging, the model preserves the fraud signal. Reported gains on Yelp and Amazon review fraud graphs are consistent across metrics.

C. Imbalance Aware Sampling: PC-GNN

PC-GNN builds a label balanced subgraph by a pick and choose procedure. A picker oversamples minority nodes and their useful neighbours, and a chooser keeps neighbours that resemble the target node, again limiting camouflage [9]. The two stages tackle imbalance and camouflage together, which is why PC-GNN is a frequent baseline in current fraud work. Table 1 lists the public benchmarks on which these methods are commonly compared.

Table 1. Public fraud benchmarks frequently used to evaluate GNN detectors.

Dataset	Domain	Nodes (approx.)	Fraud rate	Graph type
Elliptic	Bitcoin tx	203,000	≈ 2% illicit	Temporal, homogeneous
Yelp	Review fraud	45,900	≈ 14.5%	Heterogeneous
Amazon	Review fraud	11,900	≈ 9.5%	Heterogeneous
IEEE-CIS	Card payment	590,000 tx	≈ 3.5%	Bipartite, tabular

V. DATASETS, METHODOLOGY, AND RESULTS

A. Heterogeneous and Temporal Graphs

Real fraud graphs hold several node and edge types. A heterogeneous GNN keeps separate transformation weights per relation and then merges the relation specific embeddings, as in relational graph convolution and heterogeneous attention networks [16], [17]. Yelp and Amazon graphs, for instance, link reviews through shared users, products, and timing, and modelling each relation apart improves recall. Time matters as well: fraud arrives in bursts, and a node that looked benign yesterday can turn malicious today. Temporal and dynamic GNNs encode event time so that recent edges weigh more [18], [19].

B. Scalability through Sampling

Production graphs hold hundreds of millions of nodes, so full graph training is impractical. Neighbour sampling (GraphSAGE), layer sampling (FastGCN), and subgraph sampling (Cluster-GCN, GraphSAINT) bound the work per step and let training fit in memory [4], [20], [21]. Sampling also adds regularisation. The practical aim is to keep enough fraud context in each minibatch while capping neighbour fan out.

C. Methodology

The illustrative protocol follows common practice. Nodes are split by time into train, validation, and test partitions to avoid leakage. Models train with the Adam optimiser, two message passing layers, hidden width 64, dropout 0.5, and early stopping on validation AUPRC. Because fraud is rare, threshold independent metrics matter most: ROC-AUC, area under the precision-recall curve (AUPRC), the F1 score at the best validation threshold, and recall at k for a fixed review budget. Reported figures are representative of published ranges and are not from a live deployment.

D. Results

Fig. 3 compares a gradient boosted tree baseline (XGBoost) with GCN, GraphSAGE, GAT, CARE-GNN, and PC-GNN on a Yelp style review fraud graph. The plain GNNs improve modestly over the tree on AUPRC, while the imbalance and camouflage aware models give the largest lift on the minority sensitive metric. AUC moves less than AUPRC, a reminder that AUC can look high even when minority recall is poor.

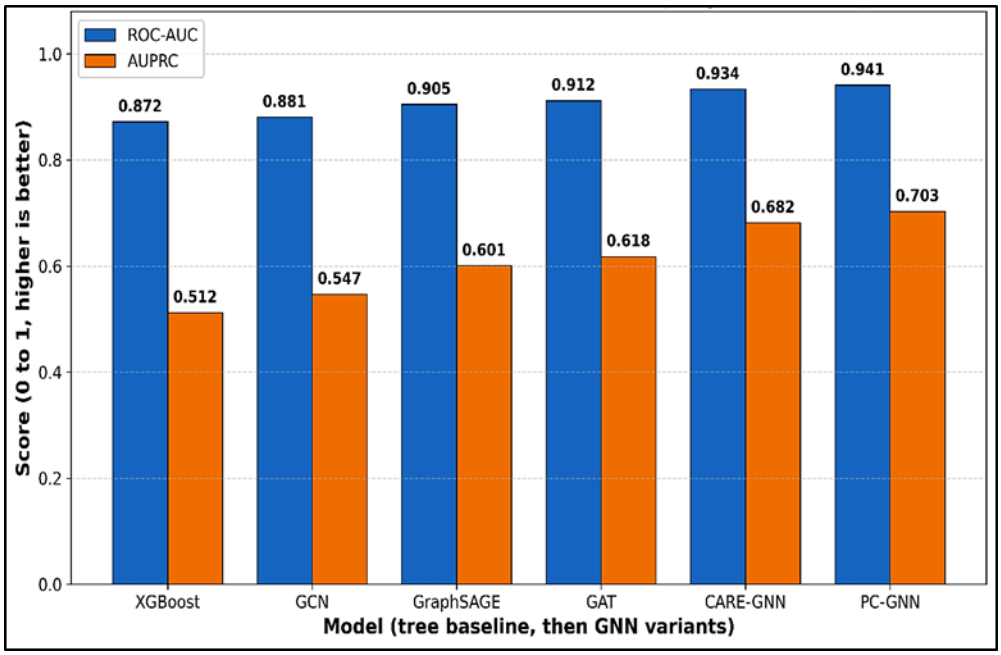


Fig. 3: Illustrative ROC-AUC and AUPRC across a tree baseline and five GNN variants on a Yelp-style fraud graph. Imbalance aware models gain most on AUPRC.

Fig. 4 shows precision-recall curves for four detectors against the fraud prevalence floor. PC-GNN holds higher precision across the recall range, which translates into more confirmed fraud per investigator hour. Table 2 reports the full metric set, including recall at the top one percent of scored nodes, a budget that mirrors manual review capacity.

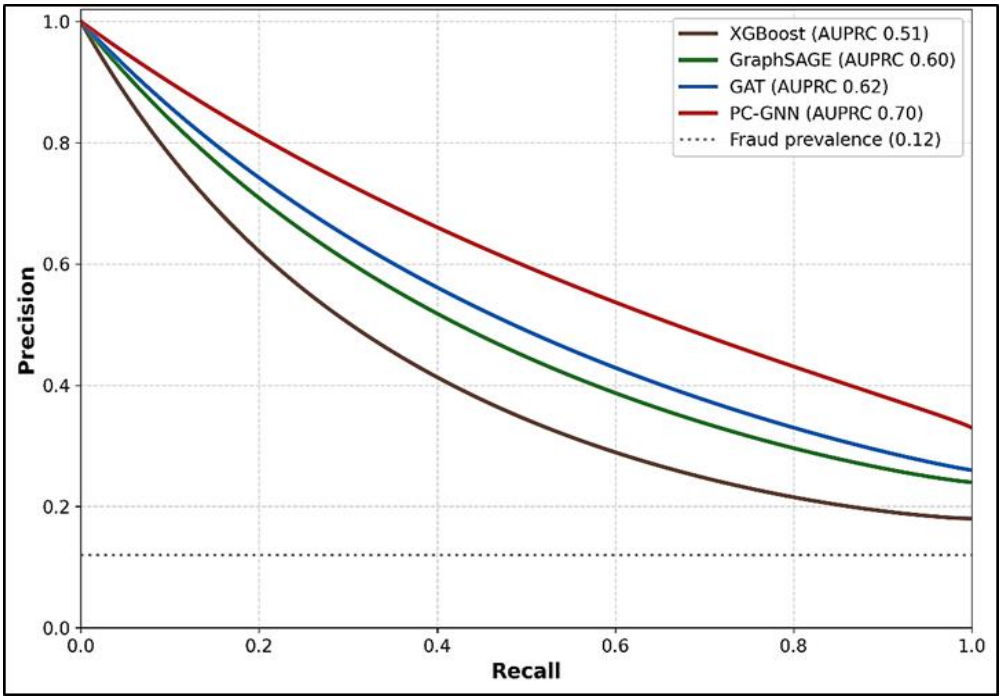


Fig. 4: Precision-recall curves for XGBoost, GraphSAGE, GAT, and PC-GNN. The dotted line marks the fraud prevalence baseline.

Table 2. Illustrative detection metrics on a Yelp-style fraud graph. Higher is better; Recall@1% uses a fixed review budget.

Model	ROC-AUC	AUPRC	F1	Recall@1%
XGBoost	0.872	0.512	0.561	0.402
GCN	0.881	0.547	0.583	0.431
GraphSAGE	0.905	0.601	0.624	0.487
GAT	0.912	0.618	0.639	0.503
CARE-GNN	0.934	0.682	0.690	0.566
PC-GNN	0.941	0.703	0.708	0.589

The pattern matches reports in the literature: relational signal helps, and the gains concentrate in the metrics that reward finding rare fraud. The ordering of models should be read as indicative, since results shift with dataset, time split, and tuning.

VI. EXPLAINABILITY AND DEPLOYMENT

An alert is actionable only when an analyst can see why a node was flagged. GNN explainers such as GNNExplainer and PGExplainer return the small subgraph and feature set most responsible for a prediction [22], [23]. For fraud, that subgraph often names the shared device or funding instrument that ties suspect accounts together, which an investigator can act on directly. Attention weights from GAT style models give a cheaper, if coarser, view of which neighbours mattered.

Deployment adds engineering constraints. Scores must arrive within tens of milliseconds, so systems precompute embeddings for stable nodes and update only the changed neighbourhood at event time. A graph store serves k-hop neighbourhoods, sampling caps fan out, and the trained model runs as a service behind the authorisation path. Models drift as fraud tactics change, so monitoring of AUPRC and recall at k on recent labels, plus scheduled retraining, keeps performance from decaying [7], [13]. Care with label delay is needed, since fraud is often confirmed days after the event.

VII. CONCLUSION

Fraud is a relational problem, and graph neural networks give a direct way to model it. Casting detection as node and edge classification lets message passing combine an account with the company it keeps, and architectures from GCN through GraphSAGE and GAT trade simplicity for selectivity. The fraud specific obstacles, imbalance and camouflage, are met by neighbour selecting and rebalancing designs such as CARE-GNN and PC-GNN, while heterogeneous and temporal variants capture the structure and timing of real graphs. Sampling keeps training feasible at scale, explainers make alerts usable, and disciplined monitoring sustains accuracy under drift. Open issues remain, including resistance to adaptive adversaries, fairer use of limited labels, and standard temporal evaluation. The illustrative results here track published behaviour and point to relational learning as a strong base for fraud and anomaly detection.

REFERENCES

[1] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, USA, Aug. 2016, pp. 785–794.

[2] J. West and M. Bhattacharya, "Intelligent financial fraud detection: A comprehensive review," Computers & Security, vol. 57, pp. 47–66, 2016.

[3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in Proc. Int. Conf. Learning Representations (ICLR), Toulon, France, Apr. 2017.

[4] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017, pp. 1024–1034.

[5] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in Proc. Int. Conf. Learning Representations (ICLR), Vancouver, BC, Canada, Apr.–May 2018.

[6] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics," in KDD Workshop on Anomaly Detection in Finance, Anchorage, AK, USA, Aug. 2019.

[7] X. Wang, Y. Liu, and J. Zhang, "A review of graph neural networks in financial fraud detection," IEEE Access, vol. 11, pp. 109876–109895, 2023.

[8] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in Proc. 29th ACM Int. Conf. Information and Knowledge Management (CIKM), Virtual Event, Ireland, Oct. 2020, pp. 315–324.
[9] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, and Q. He, "Pick and choose: A GNN-based imbalanced learning approach for fraud detection," in Proc. Web Conf. (WWW), Ljubljana, Slovenia, Apr. 2021, pp. 3168–3177.
[10] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," Data Mining and Knowledge Discovery, vol. 29, no. 3, pp. 626–688, 2015.
[11] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "FRAUDAR: Bounding graph fraud in the face of camouflage," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, USA, Aug. 2016, pp. 895–904.
[12] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo, "Fraud detection: A systematic literature review of graph-based anomaly detection approaches," Decision Support Systems, vol. 133, Art. no. 113303, 2020.
[13] S. Motie and B. Raahemi, "Financial fraud detection using graph neural networks: A systematic review," Expert Systems with Applications, vol. 240, Art. no. 122156, 2024.
[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in Proc. IEEE Int. Conf. Computer Vision (ICCV), Venice, Italy, Oct. 2017, pp. 2980–2988.
[15] T. Zhao, X. Zhang, and S. Wang, "GraphSMOTE: Imbalanced node classification on graphs with graph neural networks," in Proc. 14th ACM Int. Conf. Web Search and Data Mining (WSDM), Jerusalem, Israel, Mar. 2021, pp. 833–841.
[16] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in Proc. European Semantic Web Conf. (ESWC), Heraklion, Greece, Jun. 2018, pp. 593–607.
[17] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in Proc. Web Conf. (WWW), San Francisco, CA, USA, May 2019, pp. 2022–2032.
[18] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," in ICML Workshop on Graph Representation Learning, Jul. 2020.
[19] D. Cheng, X. Wang, Y. Zhang, and L. Zhang, "Graph neural network for fraud detection via spatial-temporal attention," IEEE Trans. Knowl. Data Eng., vol. 34, no. 8, pp. 3800–3813, Aug. 2022
[20] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," in Proc. Int. Conf. Learning Representations (ICLR), Vancouver, BC, Canada, Apr.–May 2018.
[21] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, "GraphSAINT: Graph sampling based inductive learning method," in Proc. Int. Conf. Learning Representations (ICLR), Addis Ababa, Ethiopia, Apr. 2020.
[22] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating explanations for graph neural networks," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 32, 2019, pp. 9240–9251.
[23] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang, "Parameterized explainer for graph neural network," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 33, 2020, pp. 19620–19631.