Subgraph Patterns Enhanced Graph Neural Network for Fraud Detection

Yao Zou[1], Sheng Xiang[2], Qijun Miao[1], Dawei Cheng[1,3](✉), and Changjun Jiang[1,3]

[1] Department of Computer Science and Technology, Tongji University, Shanghai, China
{sky_zy,dcheng,cjjiang}@tongji.edu.cn
[2] AAII, University of Technology Sydney, Sydney, Australia
sheng.xiang@student.uts.edu.au
[3] Shanghai Artifcial Intelligence Laboratory, Shanghai, China

Abstract. Credit card fraud is a serious problem that causes significant losses for cardholders and issuing banks. Current detection methods often focus on spatial-temporal anomalies in transactions but tend to overlook the disguising techniques of fraudsters. We notice that covert credit card fraud activities frequently manifest localized clustering characteristics, which are particularly evident in different subgraph patterns. To address this, we propose Subgraph Patterns enhanced Graph Neural Network (SPGNN), a novel approach that effectively harnesses specific traits to significantly improve fraud detection capabilities. This method employs subgraph pattern features to more sharply distinguish between fraudulent and legitimate nodes, aiding in the identification of disguised fraudsters. In particular, we devise a probabilistic neighbor selector to assist nodes in selecting more similar minority class nodes, effectively balancing data distribution and filtering out disguised nodes. Furthermore, we introduce a reinforcement learning module for supervised similarity measurement, further filtering out disguised fraudsters. Extensive experiments on several benchmark datasets demonstrate that SPGNN surpasses state-of-the-art models in detecting fraudulent activities, achieving the most advanced performance.

Keywords: Graph Neural Networks · Fraud Detection · Reinforcement Learning · subgraph patterns

1 Introduction

Financial fraud not only damages the financial well-being of individuals and businesses but also has a significant impact on the broader economy, erodes trust within the financial system, and disrupts the legal environment of society. Credit card fraud detection, a pivotal area of research, involves unauthorized fund usage, often via credit or debit cards [1]. Global card fraud losses are projected to reach $397.40 billion over the next decade [15], highlighting

M. Onizuka et al. (Eds.): DASFAA 2024, LNCS 14855, pp. 375–384, 2024.
https://doi.org/10.1007/978-981-97-5572-1_26

its significant impact on financial markets. Various models, from rules-based to machine learning approaches, have been developed to tackle fraudulent transactions. Deep learning models [4,21], such as graph neural networks (GNNs), have emerged to capture abnormal patterns as they can identify sophisticated and covert transactions by analyzing relational graphs, leading to more accurate fraud detection [23].

However, existing fraudsters engage in various behaviors to disguise themselves and evade detection [3,6,7]. These disguising actions include feature disguise and relationship disguise. 1) Feature disguise involves fraudsters mimicking the transaction characteristics of legitimate entities, such as transaction frequency and preferences. 2) Relationship disguise involves engaging with many legitimate entities to dilute node information and avoiding direct transactions with other fraudsters to evade detection as part of a fraud ring, which is insidious as it prevents the aggregation and transmission of valuable information.

Some recent work has noticed similar challenges. PCGNN [11] reduces redundant edges to tackle disguise with benign entities. Liu et al. [12] identify feature-disguising fraudsters with a score and merge context embeddings. Yu et al. [19] decompose multi-layered tree subgraphs to alleviate information dilution. However, while the aforementioned methods address some fraudulent behavior aspects, they're inadequate when fraudsters intentionally avoid direct transactions. When fraudulent nodes are amidst benign entities, messages aren't effectively transmitted, rendering methods ineffective. This underscores a critical gap in current methodologies, where indirect transactional behaviors are insufficiently considered, potentially overlooking sophisticated fraudulent activities.

We have noticed that covert credit card fraud activities frequently manifest localized clustering characteristics, which are particularly evident in different subgraph patterns. This distinction remains evident even in scenarios where fraudsters avoid direct transactions. Therefore, we propose Subgraph Patterns enhanced Graph Neural Network (SPGNN) to effectively tackle the challenge posed by disguise behaviors. Our method focuses on subgraph patterns, particularly targeting the disguise of avoiding direct transactions. Additionally, we introduce a probabilistic neighbor selector that integrates label distribution and node similarity to address other disguise issues. We also incorporate a reinforcement learning (RL) module into the training phase. This module employs a supervised similarity metric to further filter disguised fraudsters and guide the adaptive exploration of the optimal neighbor threshold. The contributions of this paper are summarized as follows:

1) To the best of our knowledge, this is the first work that leverages the power of subgraph patterns, which have the ability to effectively characterize and discriminate fraudsters to address the challenge of disguised fraudsters in credit card fraud detection.
2) We designed a probabilistic neighbor selector as the first layer of filtering for feature disguise and relationship disguise connected to multiple benign entities. Additionally, we employ an RL module for supervised similarity measurement, further filtering out disguised fraudsters.

3) We conduct extensive experiments to compare our method with state-of-the-art baselines on both public and real-world datasets. The results demonstrate that our model significantly improves GNN performance on graphs containing disguised fraudsters.

2 Preliminaries

2.1 Subgraph Patterns Analysis

Figure 1 visualizes scaled subgraph pattern features, with the left section depicting fraudulent transactions and the right section showing legitimate ones. Detailed subgraph pattern extraction steps will be described later. Subgraph patterns of 2–3 nodes are shown on the right, with the central node represented by X. Neighboring nodes are labeled as 1 for fraudulent and 0 for benign, resulting in features like X-0. Statistical analysis found that most triangle subgraph patterns resulted in zeros for both fraudulent and benign nodes. There are significant differences in the color distribution of the feature heatmaps between fraudulent and benign nodes. In the case of fraudulent transactions, certain subgraph pattern features (such as X-0-1) exhibit more pronounced color blocks in fraud features, indicating that these subgraph pattern feature values are highly significant in fraudulent activities. We opt for subgraph pattern features of 2–3 nodes (excluding triangle subgraph patterns), denoted as $\mathcal{G}_{sub} = \{\mathcal{G}_1, \mathcal{G}_2 \cdots \mathcal{G}_n\}$.

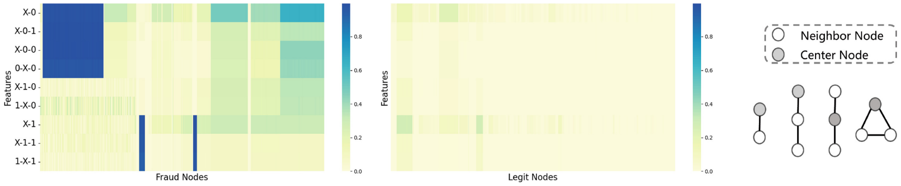

Fig. 1. Heat maps of subgraph pattern features from both fraudulent and legitimate transactions. The y-coordinate represents the corresponding subgraph pattern features. The central node is indicated by X, while neighboring nodes are denoted as 1 for fraudulent neighbors and 0 for benign neighbors.

2.2 Problem Definition

Definition 1. Transactions. A transaction record r can be defined as an attribute tuple $r = \{s, t, \boldsymbol{x}\}$ in the transaction payment process, where s denotes the transaction initiator, t represents the merchant or receiver, and $\boldsymbol{x}$ means the characteristic vector of the transaction.

Definition 2. Graph & Subgraph Pattern. A graph is an ordered pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ is a set, whose items are called vertices or nodes, and $\mathcal{E}$ is a set of unordered pairs of vertices. A graph $\mathcal{G}^{'} = (\mathcal{V}^{'}, \mathcal{E}^{'})$ is a subgraph pattern of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}^{'} \subseteq \mathcal{V}$, $\mathcal{E}^{'} \subseteq \mathcal{E}$, and $(v_1, v_2) \in \mathcal{E}^{'} \rightarrow v_1, v_2 \in \mathcal{V}^{'}$.

Problem Definition. Given the original transaction records $\mathcal{R} = \{\mathcal{S}, \mathcal{T}, \boldsymbol{\mathcal{X}}\}$, we construct a transaction graph $\mathcal{G} = (\mathcal{V}, \boldsymbol{\mathcal{X}}, \mathcal{E}, \mathcal{Y})$. Firstly, we treat each transaction as a discrete node in the graph, $\mathcal{V} = \{v_1, \cdots, v_n\}$, $\mathcal{E} = \{\emptyset\}$. Each node v_i has a d-dimensional feature vector $\boldsymbol{x}_i \in \mathbb{R}^d$ and $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ represents a set of all node features. When the merchants or transaction initiators of two transactions v_i, v_j are the same, we add an edge between their corresponding nodes, $\mathcal{E} = \mathcal{E} \cup \{e_{i,j}\}$. $\mathcal{Y} = \{y_1, \cdots, y_n\}$, $y_i \in \{0, 1\}$ denotes the label of node v_i. If a transaction is reported by a cardholder or identified by financial experts as fraudulent, we label it as 1; otherwise, it is labeled as 0. For the set of subgraph patterns $\mathcal{G}_{sub} = \{\mathcal{G}_1, \mathcal{G}_2 \cdots \mathcal{G}_n\}$, subgraph pattern feature extraction is performed to obtain the subgraph pattern features which will be combined with the $\boldsymbol{\mathcal{X}}$ to form a new feature $\boldsymbol{\mathcal{X}}^{'}$. We hope to infer the possibility of fraudulent transactions based $\mathcal{G}^{'} = (\mathcal{V}, \boldsymbol{\mathcal{X}}^{'}, \mathcal{E}, \mathcal{Y})$.

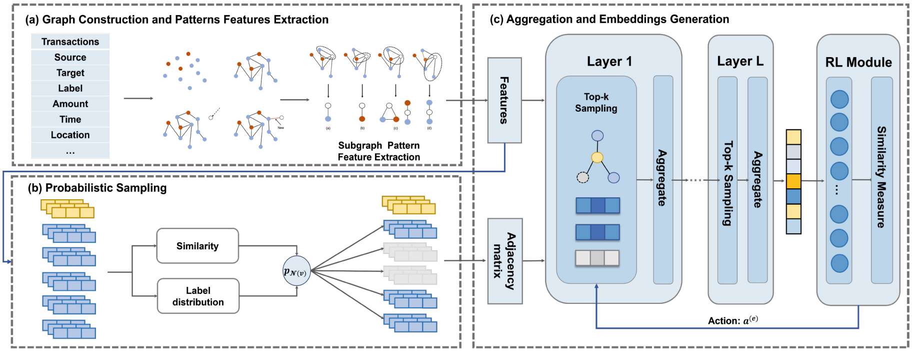

Fig. 2. The illustration of the proposed SPGNN.

3 The Proposed Approaches

3.1 Model Architecture

Figure 2 shows the overall architecture of SPGNN. We first generate complex transaction graphs based on original transaction records. Subsequently, subgraph pattern feature extraction is performed to obtain the subgraph pattern features, which are then combined with the original node attribute features. We implement a probabilistic neighbor selector that takes into account label distribution and node similarity to improve neighbor selection. This approach focuses on minority class nodes with high similarity, thereby achieving a balanced data distribution and effective identification of concealed nodes. Subsequently, these refined node features are fed into the GNN. Through the integration of multiple neighbor selectors and aggregation layers, the GNN effectively shares and compiles information across nodes. Simultaneously, the training process of the GNN serves as the environment for the RL module, in which the RL agent, incorporating supervised similarity measurement, further filters out disguised fraudsters.

3.2 Probabilistic Neighbor Selector and RL Module

We devise a probabilistic neighbor selector to pick more informative nodes to filter disguised fraudsters. The key idea lies in incorporating label distribution information and node similarity into the sampling process. For the neighbor selector, those minority class nodes with high similarity have a higher sampling probability. The sampling probability of neighbor node u can be defined:

$$p_u = \frac{S\left(v,u\right)}{\sum_{v' \in \mathcal{N}(v)} S\left(v,v'\right) \cdot DF(v, C(u))}, \tag{1}$$

where v is the central node and $\mathcal{N}(v)$ represents the neighbor set of node v. $DF(v, (C(u)))$ denotes the label frequency of class $C(u)$ in $\mathcal{N}(v)$. $S\left(v,u\right)$ is the similarity between neighbor node u and the center node v (We set it as cosine similarity). To refine fraud detection, we integrate an RL module in training. It learns an adaptive parameter p, acting as a threshold, ensuring nodes aggregate only with similar neighboring nodes. Inspired by LAGCN [2], we incorporate a Single-layer Perceptron (SLP) as a new layer before the aggregation layer of GNN. This SLP predicts node labels based on their embeddings. For a center node v at the l-th layer and edge $(v, v') \in \mathcal{E}$, the distance between v and v' is defined as:

$$\mathcal{D}ist^{(l)}(v,v') = \left|\sigma\left(\mathrm{SLP}^{(l)}(\mathbf{h}_v^{(l-1)})\right) - \sigma\left(\mathrm{SLP}^{(l)}(\mathbf{h}_{v'}^{(l-1)})\right)\right|, \tag{2}$$

where $\mathbf{h}_v^{(l)}$ is the hidden embedding at the l-th layer of node v, $\mathbf{h}_v^{(0)} = \boldsymbol{x}_v$ is the input feature, and σ is a nonlinear activation function (we use sigmoid). The similarity measure is defined as:

$$\mathcal{S}imi^{(l)}(v,v') = 1 - \mathcal{D}ist^{(l)}(v,v'). \tag{3}$$

To optimize computational efficiency, we simplify the input by using only the node embedding, rather than combined embeddings as in LAGCN [2]. This reduces the time complexity of the proposed similarity measure from $O(|\mathcal{V}|\bar{D}d)$ to $O(|\mathcal{V}|d)$. We define the cross-entropy loss of the SLP at the l-th layer as:

$$\mathcal{L}_{\mathrm{Simi}}^{(l)} = \sum_{v \in \mathcal{V}} -\log\left(y_v \cdot \sigma\left(\mathrm{SLP}^{(l)}\left(\mathbf{h}_v^{(l)}\right)\right)\right), \tag{4}$$

where σ is the sigmoid activation function. We define the RL process as a Bernoulli Multi-armed Bandit (BMAB) $\mathcal{B}(A, R, T)$ between the neighbor selector and the GNN with the similarity measure. Here, A is the action space, R is the reward function, and T is the terminal condition.

Reward function. Optimal p minimizes distance (or maximizes similarity) between the central node and its neighbors in the first layer. Binary stochastic reward is based on average distance differences between consecutive epochs. Average neighbor distances for epoch e are calculated as:

$$O\left(\mathcal{D}ist\right)^{(e)} = \frac{\sum_{v \in \mathcal{V}_{\mathrm{train}}} \mathcal{D}ist^{(1)}(v,v')^{(e)}}{|\mathcal{V}_{\mathrm{train}}\ |}. \tag{5}$$

denoted as $O^{(e)}$ for simplicity. The reward function for epoch e is defined as:

$$a^{(e)} = \begin{cases} +1, O^{(e-1)} - O^{(e)} \geq 0, \\ -1, O^{(e-1)} - O^{(e)} < 0. \end{cases} \tag{6}$$

Action: p adjusts incrementally based on the reward (Eq. (7)), aiming to reduce neighboring nodes when distance increases.

$$p = p + a^{(e)} \cdot \tau \tag{7}$$

Terminal: Eq. (8) defines a terminal condition, signifying RL module convergence in recent epochs, identifying an optimal threshold p.

$$\left| \sum_{e-10}^{e} a^{(e)} \right| \leq 1, \text{ where } e \geq 10. \tag{8}$$

3.3 Aggregation and Embedding Generation

We choose the mean operator as the aggregator of GNN, represented as:

$$\mathbf{h}_v^{(l)} = \sigma \left(W^{(l)} \cdot \text{MEAN} \left(\left\{ \mathbf{h}_v^{(l-1)} \right\} \cup \left\{ \mathbf{h}_u^{(l-1)}, \forall u \in \mathcal{N}(v) \right\} \right) \right), \tag{9}$$

where the $W^{(l)}$ is the l-th parameter matrix. Following the aggregation step, an MLP classifier is trained together with GNNs to minimize the cross-entropy loss.

$$p_v = \text{MLP}(\mathbf{h}_v^{(L)}), \tag{10}$$

where L is the number of layers. We define the graph-based loss function:

$$\mathcal{L}_{\text{gnn}} = -\sum_{v \in \mathcal{V}} \left[y_v \log p_v + (1 - y_v) \log (1 - p_v) \right] \tag{11}$$

Combined with the similarity measure loss, the loss of SPGNN is defined as:

$$\mathcal{L}_{\text{SPGNN}} = \lambda \cdot \mathcal{L}_{\text{Simi}}^{(1)} + \mathcal{L}_{\text{gnn}}, \tag{12}$$

where λ is a weighting parameter.

4 Experiments

4.1 Experiments Settings

Datasets. We collected real-world credit card transaction data from a major commercial bank spanning January 1 to December 31, 2021. The dataset, referred to as **CCDS** (**C**redit **C**ard Fraud Detection **D**ata**S**et), consists of 140,576 transactions involving 20,313 unique users. Transaction features include location, amount, and type. Additionally, we utilized two public datasets for experiments: the YelpChi graph dataset containing hotel and restaurant reviews and the Amazon graph dataset comprising product reviews of musical instruments. Basic statistics for the datasets are summarized in Table 2.

Compared Methods and Settings. We compare SPGNN with the baselines: GCN [8], GEM [10], GAT [16], FdGars [18], Play2Vec [20], GeniePath [13], Graphconsis [12], SEMIGNN [17], CAREGNN [5], PCGNN [11]. Sub-models of SPGNN are SPGNN-*nosample/nosubgragh/noRL*, in which probabilistic neighbor selector, subgraph pattern features, or RL module are not used. SPGNN is the full proposed model. Baseline methods utilize their originally proposed optimal parameters. SPGNN employs a two-layer aggregation scheme with an embedding size of 64. We use a learning rate of 0.01 and an RL step size of 0.01. Evaluation includes AUC and F1 metrics on three datasets.

Table 1. Fraud detection performance on three datasets.

Model	YelpChi		Amazon		CCDS	
	AUC	F1	AUC	F1	AUC	F1
GCN	0.5310	0.4614	0.5305	0.4413	0.5266	0.4175
GEM	0.5201	0.5017	0.5292	0.4983	0.5388	0.6599
GAT	0.5322	0.4634	0.5357	0.4822	0.5333	0.6624
FdGarS	0.5133	0.4304	0.6601	0.3757	0.5475	0.3689
Play2Vec	0.5231	0.4606	0.5205	0.4587	0.5380	0.5022
GeniePath	0.6761	0.5915	0.7832	0.7952	0.6150	0.6234
Graphconsis	0.7060	0.6041	0.8225	0.7766	0.5893	0.6402
SEMIGNN	0.5201	0.1045	0.8782	0.7819	0.5473	0.4485
CAREGNN	0.7934	0.6493	0.9115	0.8531	0.6534	0.5771
PCGNN	0.7987	0.6300	0.9405	0.8865	0.6795	0.6077
SPGNN-Nosample	0.7793	0.5796	0.9362	0.8929	0.7209	0.6222
SPGNN-Nosubgraph	0.7579	0.5897	0.9217	0.8971	0.6922	0.6467
SPGNN-NoRL	0.7825	0.5165	0.9513	0.8923	0.7189	0.6284
SPGNN-all	**0.8013**	**0.6517**	**0.9519**	**0.9159**	**0.7352**	**0.6654**

4.2 Fraud Detection Performance

We repeated the experiments ten times for each method and have shown the average performance in Table 1. The first five rows of Table 1 report the results of some classic graph-based methods, including GCN, GEM, GAT, FdGars, and Play2Vec. It is clear that the results of GCN and GEM are not satisfactory, demonstrating the limitations of shallow models in addressing complex fraud patterns. GAT introduces the attention mechanism to improve the indiscriminate aggregation of neighbor information compared with the first two methods, but the improvement is not significant. FdGars and Play2Vec improve performance, partially due to their enlarged model capacities. GeniePath and Graphconsis perform closely to each other and better than the first five methods. The results demonstrate the effectiveness of aggregating neighborhoods based on node similarity in detecting fraud transactions. SEMIGNN performs well in the Amazon dataset, while it performs poorly on the Yelp and CCDS datasets as it is an unsupervised method and data distribution can seriously affect its model performance. CAREGNN introduces RL module, resulting in improved performance. PCGNN achieved more competitive results by sampling nodes and edges.

Lines 9-12 show the results of SPGNN and its sub-models. It should be noted that SPGNN-*nosubgraph* performs considerably lower than the other two submodels, which proves the superior performance of the subgraph pattern features. In general, SPGNN performed the best across all metrics.

Table 2. Node similarity statistics.

Dataset	YelpChi	Amazon	CCDS
Node	45,954	11,948	140,576
Edge (M)	3.85	4.40	19.96
Fraud (%)	14.5	9.5	7.8
Old	0.77	0.65	0.69
New	0.55	0.46	0.57

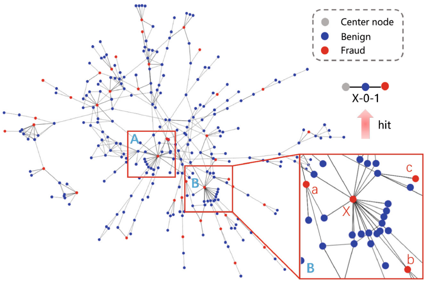

Fig. 3. The layout of a typical graph in Amazon.

4.3 Interpretative Analysis

We calculate the feature similarity of neighboring nodes based on the Euclidean distance of their feature vectors, ranging from 0 to 1. The average similarity is normalized with respect to the total number of edges, as shown in Table 2. Initially, the feature similarity was unusually high. However, after incorporating subgraph pattern features and re-measuring, we observed a decrease in similarity of at least 13.2%, indicating significant differences between fraudulent and legitimate nodes' subgraph pattern features, thus enhancing fraud detection accuracy. Figure 3 illustrates a typical case in the Amazon dataset, involving 296 users. Fraudulent nodes are marked in red, while normal nodes are in blue. Box A showcases the first type of relationship disguise: engaging with many legitimate entities. Box B illustrates the second type: avoiding direct transactions. Zooming in on box B, it shows a central node labeled X, surrounded by benign entities and not directly connected to other fraudulent nodes. However, this central node exists in a special subgraph pattern X-0-1 with nodes a, b, and c. This further demonstrates how subgraph pattern features aid in identifying disguised fraudsters even without direct connections.

5 Related Work

Credit card fraud detection has been a significant application area for machine learning techniques. Traditional methods often rely on time perception [22] to capture fraudulent transaction patterns. However, these approaches may overlook relevant characteristics or focus solely on temporal features. Recently, GNNs have shown promise in fraud detection [14], yet they struggle to effectively identify disguised fraudsters. To address this challenge, existing sampling methods are introduced [9]. Despite their advancements, these methods often suffer from high computational complexity and may struggle with complex disguise scenarios. In contrast, SPGNN combines a simpler similarity measure with lower computational complexity and leverages subgraph pattern features to effectively identify disguised fraudsters, enhancing the capabilities of GNNs.

6 Conclusion

This paper addresses credit card fraud detection, a critical real-world challenge. Recognizing the significant impact of fraudsters' disguise tactics on GNN-based detectors, we propose a probabilistic neighbor selector for initial disguise filtering. Furthermore, we integrate reinforcement learning to adjust neighbor selection based on node similarity, enhancing detection accuracy. Leveraging subgraph pattern features, we unveil intricate transaction dependencies, capturing complex interactions in the graph and offering insights into fraudulent activities. SPGNN's capability to identify and utilize subgraph patterns is pivotal for precise fraud detection, minimizing false positives. Extensive experiments across various fraud datasets validate SPGNN's superiority over baseline methods, underscoring its efficacy and practical applicability in real-world scenarios.

Acknowledgments. This work was supported by the National Key R&D Program of China (Grant no. 2022YFB4501704), the National Natural Science Foundation of China (Grant no. 62102287), and the Shanghai Science and Technology Innovation Action Plan Project (Grant no. 22YS1400600 and 22511100700).

References

1. Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J.C.: Data mining for credit card fraud: a comparative study. Decis. Support Syst. **50**(3), 602–613 (2011)
2. Chen, H., Wang, L., Wang, S., Luo, D., Huang, W., Li, Z.: Label aware graph convolutional network - not all edges deserve your attention. CoRR abs/1907.04707 (2019)
3. Cheng, D., Chen, C., Wang, X., Xiang, S.: Efficient top-k vulnerable nodes detection in uncertain graphs. IEEE Trans. Knowl. Data Eng. **35**(2), 1460–1472 (2021)
4. Cheng, D., Wang, X., Zhang, Y., Zhang, L.: Risk guarantee prediction in networked-loans. In: IJCAI (2020)
5. Dou, Y., Liu, Z., Sun, L., Deng, Y., Peng, H., Yu, P.S.: Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In: CIKM, pp. 315–324 (2020)

6. Dou, Y., Ma, G., Yu, P.S., Xie, S.: Robust spammer detection by nash reinforcement learning. In: SIGKDD, pp. 924–933. Association for Computing Machinery, New York (2020)
7. Kaghazgaran, P., Caverlee, J., Squicciarini, A.: Combating crowdsourced review manipulators: a neighborhood-based approach. In: WSDM, pp. 306–314 (2018)
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
9. Li, F., Zhang, T., Cui, S., Liu, H., Ren, Z., Di, D., Wang, X., Zhang, P., Yu, G.: A sampling method based on forecasting and combinatorial optimization for high performance a/b testing. Front. Comp. Sci. **17**(6), 176616 (2023)
10. Lin, W., Lan, H., Li, B.: Generative causal explanations for graph neural networks. In: ICML, pp. 6666–6679 (2021)
11. Liu, Y., Ao, X., Qin, Z., Chi, J., Feng, J., Yang, H., He, Q.: Pick and choose: a gnn-based imbalanced learning approach for fraud detection. In: WWW, pp. 3168–3177 (2021)
12. Liu, Z., Dou, Y., Yu, P.S., Deng, Y., Peng, H.: Alleviating the inconsistency problem of applying graph neural network to fraud detection. In: SIGIR, pp. 1569–1572 (2020)
13. Liu, Z., Chen, C., Li, L., Zhou, J., Li, X., Song, L., Qi, Y.: Geniepath: graph neural networks with adaptive receptive paths. In: AAAI, vol. 33, pp. 4424–4431 (2019)
14. Ma, J., et al.: Fighting against organized fraudsters using risk diffusion-based parallel graph neural network. In: IJCAI, pp. 6138–6146 (2023)
15. Nilson Report: Card fraud losses (2022). https://nilsonreport.com/research/research-14th-edition/. Accessed 20 Dec 2023
16. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: ICLR (2018)
17. Wang, D., e al.: A semi-supervised graph attentive network for financial fraud detection. In: ICDM, pp. 598–607. IEEE (2019)
18. Wang, J., Wen, R., Wu, C., Huang, Y., Xiong, J.: Fdgars: fraudster detection via graph convolutional networks in online app review system. In: WWW, pp. 310–316 (2019)
19. Wang, Y., Derr, T.: Tree decomposed graph neural network. In: CIKM, pp. 2040–2049 (2021)
20. Wang, Z., Long, C., Cong, G., Ju, C.: Effective and efficient sports play retrieval with deep representation learning. In: SIGKDD, pp. 499–509 (2019)
21. Xiang, S., et al.: Semi-supervised credit card fraud detection via attribute-driven graph representation. In: AAAI, vol. 37, pp. 14557–14565 (2023)
22. Xie, Y., Liu, G., Yan, C., Jiang, C., Zhou, M., Li, M.: Learning transactional behavioral representations for credit card fraud detection. TNNLS (2022)
23. Zhang, R., et al.: Pre-trained online contrastive learning for insurance fraud detection. In: AAAI, pp. 22511–22519 (2024)