ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446523

F2GNN: AN ADAPTIVE FILTER WITH FEATURE SEGMENTATION FOR GRAPH-BASED FRAUD DETECTION

Guanghui Hu[1,2] *Yang Liu*[*2] *Qing He*[1,2] *Xiang Ao*[*1,2,3]

[1] Henan Institute of Advanced Technology, Zhengzhou University, Zhengzhou 450002, P.R. China
[2] Key Laboratory of AI Safety & Security, Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
[3] CASMINO Ltd., Suzhou 215000, China

ABSTRACT

Graph Neural Networks (GNNs) have received remarkable success in identifying fraudulent activities on graphs. Most approaches leverage the full user feature together and aggregate the messages from its neighbors by a graph filter. However, due to the adversarial activities like the camouflage of fraudsters, most dimensions of fraudsters' features resemble normal users, and modeling the features as a whole cannot fully explore the small-portion fraudulent features. In this paper, we attempt to segment the user features and apply adaptive graph filters on each segmentation for better modeling of fraudulent features. We propose an adaptive filter with feature segmentation (shortened as F^{2}GNN) to alleviate these challenges. Experimental results on two real-world datasets demonstrate that F^{2}GNN outperforms state-of-the-art baselines for graph-based fraud detection. In addition, the adaptive filter with feature segmentation can effectively address the class imbalance problem in the task of fraud detection.

***Index Terms*—** Graph neural networks, fraud detection, adaptive filter

1. INTRODUCTION

Fraud detection refers to the analysis and identification of potential fraudulent behaviors within a system, widely applied in domains such as finance [1, 2], e-commerce [3], and review management [4, 5]. In recent years, graph-based fraud detection has garnered increasing attention from both academic [4, 6] and industrial communities [1, 7]. This is because graph data reflects the interactive behaviors among users across different relationships, providing rich information for fraud detection.

Existing GNN-based fraud detection approaches [8, 9] have achieved good performance, most of them leveraging full user features and aggregating neighboring information through graph filters. However, fraudsters actively engage in behavior camouflage to evade detection. Take spammers for example, they may strategically send spam emails only on a few dates while appearing as benign users in most times. This implies that within the rich user features, the features representing fraudulent activities account for only a small portion, while fraudsters closely resemble benign users in most features. Therefore, modeling the features as a whole can easily overlook the small portion of hidden fraudulent information.

On the other hand, the number of fraudsters is usually much smaller than benign users. Fraud detection faces the challenge of class imbalance [10]. However, the vanilla GNNs are not well-suited for addressing the imbalance problem in fraud detection. Because they are essentially low-pass filters [11, 12]. They will dilute fraudsters' features when aggregating neighbor information through summation or average operations. Existing methods [6, 4] for fraud detection mitigate the impact of imbalance by selectively aggregating neighboring information, but they still operate as low-pass filters. Recent research [13] indicates that fraudsters can cause a rightward shift in the spectral energy distribution. This implies that high-frequency information also contains features related to fraudsters. Therefore, in fraud detection, it is necessary to introduce a high-pass filter module to capture high-frequency fraudulent information.

In this paper, we propose a novel model called Adaptive Filter with Feature Segmentation (F^{2}GNN for short) to address these challenges in fraud detection. The framework consists of three steps: (1) Segment node features to improve the granularity of fraud information mining. (2) Apply high-pass and low-pass adaptive filters to fully explore the segmented features. (3) Aggregate the node embeddings after adaptive filtering to update node representation. In addition, we avoid applying dropout to node features before the filtering operation and only apply once non-linear transformation to the original features. Because these operations will disrupt the already scarce fraudulent features, leading to distortion in the filtering results. We extensively evaluate our approach on two real-world fraud detection datasets. The experimental results demonstrate that our method outperforms state-of-the-art baselines with satisfactory performance. Furthermore, utilizing the adaptive filter with feature segmentation effectively mitigates the class imbalance problem.

2. PRELIMINARIES

Definition 3.1 (Multi-relation Graph). Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{X}, \{\mathcal{E}_r\}|_{r=1}^{R}, \mathcal{Y}\}$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes, n is the number of nodes; $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ is the set of node features, $x_i \in \mathbb{R}^d$ is i^{th} node feature, d is the dimension of feature; $\{\mathcal{E}_r\}$ is the set of edges with a relation $r \in \{1, \ldots, R\}$, note that an edge can be associated with multiple relations and there are R different types of relations. $\mathcal{Y}$ is the set of labels for each node in $\mathcal{V}$.

Definition 3.2 (Graph-based fraud detection). For the fraud detection problem, the node v represents the target entity, which has a label $y_v \in \{0, 1\} \in \mathcal{Y}$. The label 0 represents benign and 1 represents fraud. The graph-based fraud detection problem is a semi-supervised binary node classification problem on the graph. The trained models are then used to predict the suspiciousness of unlabeled nodes.

*Corresponding Author.

Definition 3.3 (Graph Filtering). For a graph $\mathcal{G}$, $A \in \mathbb{R}^{n\times n}$ denotes the adjacency matrix and $D_{i,i} = \sum_j A_{i,j} \in \mathbb{R}^{n\times n}$ is a diagonal degree matrix. The normalized graph Laplacian matrix is defined as $L = I - D^{-1/2}AD^{-1/2}$,where I denotes the identity matrix. Because L is a real symmetric matrix, we have $L = U\Lambda U^T$, where $\Lambda = diag([\lambda_1, \lambda_2, \ldots, \lambda_n])$. Graph Fourier transform of a signal $x \in \mathbb{R}^n$ is defined as $\widehat{x} = U^T x$ and Graph Fourier inverse transform $x = U\widehat{x}$. Graph convolution $*_G$ between the signal x signal f is:

$$f *_G x = U((U^T f) \odot (U^T x)) = U g_\theta U^T x \tag{1}$$

where $\odot$ denotes Hadamard product, g_θ is a diagonal matrix. The convolution filter in the spectral domain is $U^T f$. The graph convolution operation is also referred to as Graph Filtering.

3. METHODOLOGY

This section presents the details of our proposed method. Firstly, we provide an overview of the entire framework in Section 3.1. Then, in Section 3.2 and 3.3, we respectively introduce two key modules: feature segmentation and adaptive filter. Section 3.4 describes the message passing and aggregation process for each layer of the model. Finally, we define the model loss and training process.

3.1. Overview

The overall framework is shown as Figure 1. For instance, a node v has a single neighbor u. To obtain the representation of the target node v, there are mainly three steps: feature segmentation, adaptive filtering, and aggregation. h_v and h_u denote the embeddings of node u and v, respectively. Firstly, we segment them into S_n segments ($S_n = 2$ for instance). Then, for each corresponding segment, perform adaptive filtering on h_u and h_v separately. Finally, concatenate the embeddings from each segment to obtain the new representation of node v, denoted as h'_v.

3.2. Feature Segmentation

To fully exploit the node features information, we first perform a non-linear transformation on the original features to map them into a higher-dimensional embedding. Then divide the embedding vectors into S_n segments, the dimension of each segment is S_d. We use $h_i^{(l)}$ to represent the embedding of node i at layer l.

$$h_i^{(0)} = \sigma(W_s x_i), \tag{2}$$

where $W_s \in \mathbb{R}^{S_d S_n \times d}$ is the parameter matrix, $\sigma(.)$ is nonlinear activation function. Now $h_i^{(0)} \in \mathbb{R}^{S_d S_n}$.

$$h_i^{(0)} = \Big\Vert_{k=1}^{S_n} h_{i,k}^{(0)}, \tag{3}$$

where $k \in \{1, 2, \ldots, S_n\}$, $h_{i,k}^{(0)} \in \mathbb{R}^{S_d}$ denotes k^{th} segment of $h_i^{(0)}$, and $\Vert$ represents concatenation.

3.3. Adaptive Filter Analysis

GCN [14] is a simplified version of ChebNet [15] and can be essentially viewed as a low-pass filter $\mathcal{F}_L$. Correspondingly, there is a high-pass filter denoted as $\mathcal{F}_H$.

$$\mathcal{F}_L = \varepsilon I + D^{-1/2}AD^{-1/2}, \tag{4}$$

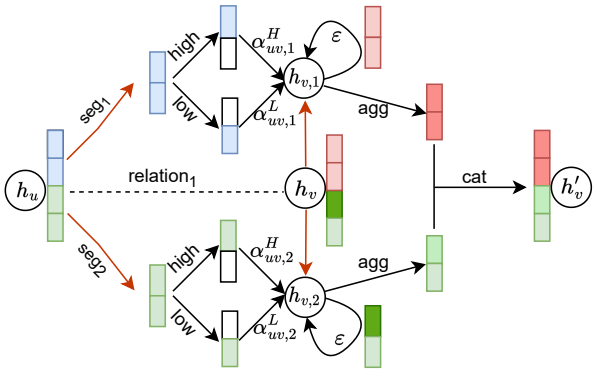


Fig. 1: The process of segmented adaptive filtering in F^{2}GNN. Node u is a neighbor of node v, and $\alpha^H_{uv,1}$ and $\alpha^L_{uv,1}$ denote the respective proportions of high-pass and low-pass filtering in the first segment.

$$\mathcal{F}_H = \varepsilon I - D^{-1/2}AD^{-1/2}, \tag{5}$$

where $\varepsilon \in [0, 1]$ is a scaling hyper-parameter. Given the input node features $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\} \in \mathbb{R}^{n\times d}$, we employ the attention mechanism to integrate low-pass and high-pass filtering.

$$h'_i = \alpha^L_{ij}(\mathcal{F}_L \cdot \mathbf{H})_i + \alpha^H_{ij}(\mathcal{F}_H \cdot \mathbf{H})_i = \varepsilon h_i + \sum_{j\in N(i)} \frac{\alpha^L_{ij} - \alpha^H_{ij}}{\sqrt{d_i d_j}} h_j, \tag{6}$$

where h'_i is the updated node representation of node i. $N(i)$ and d_i represent the neighbor set and degree of node i, respectively. α^L_{ij} and α^H_{ij} represent the proportions of low-pass and high-pass filtering, respectively. When setting $\alpha^L_{ij} + \alpha^H_{ij} = 1$ and $\alpha_{ij} = \alpha^L_{ij} - \alpha^H_{ij}$, we have $\alpha_{ij} \in [-1, 1]$ as a learnable coefficient.

When $\alpha_{ij} > 0$, the updated node i becomes closer to node j, and when $\alpha_{ij} < 0$, the updated node i diverges from node j, so α_{ij} can be learned based on the similarity between node i and node j.

3.4. Segmented Filtering and Aggregation

For each layer l, given r^{th} relation subgraph $\mathcal{G}_r = \{\mathcal{V}, \mathcal{X}, \{\mathcal{E}_r\}, \mathcal{Y}\}$, $\mathcal{E}_r$ is the edge set of graph $\mathcal{G}_r$. We use $\mathcal{N}_r(v)$ to represent the neighbor set of the node v in $\mathcal{G}_r$. For an edge $e_{u,v} \in \mathcal{E}_r$, a node v and any of its first-order neighbors $u \in \mathcal{N}_r(v)$, we calculate the learnable coefficients between node u and node v for each segment:

$$\alpha^{(l-1),r}_{uv,k} = tanh(W_t[h^{(l-1)}_{u,k} || h^{(l-1)}_{v,k}]), \tag{7}$$

where $W_t \in \mathbb{R}^{1\times 2S_d}$ is a learnable weight vector, tanh(·) is the hyperbolic tangent function. Then, we calculate the message from the first-order neighbor u to the node v.

$$msg^{(l-1)}_{uv} = \Big\Vert_{k=1}^{S_n} \frac{\alpha^{(l-1),r}_{uv,k} h^{(l-1),r}_{u,k}}{\sqrt{d_u d_v}}. \tag{8}$$

To avoid model overfitting, we apply random dropout to the message matrix [16] during the aggregation process.

$$h^{(l),r}_v = \varepsilon h^{(0)}_v + \sum_{u\in\mathcal{N}(v)} \overline{msg}^{(l-1)}_{uv}, \tag{9}$$

where $\overline{msg}^{(l-1)}_{uv}$ refers to the message vector from node u to node v after random dropout in the layer $(l-1)$, and $h^{(l),r}_v$ represents the embedding of node v in the l^{th} layer of the r^{th} relation subgraph.

Finally, for multiple relation subgraphs, we use concatenation instead of summation to combine the representations of the node v from R relation subgraphs. Transforming it into a lower-dimensional embedding, we obtain the final representation of node v.

$$h_v^{(l),all} = \mathop{\Big\|}_{r=1}^{R} h_v^{(l),r} \tag{10}$$

$$h_v^{(L)} = W_d h_v^{(l),all}, \tag{11}$$

where $W_d \in \mathbb{R}^{S_d S_n \times S_d S_n R}$ is the parameter matrix.

3.5. Training

We pass the final representation $h_v^{(L)}$ of node v through an MLP to calculate the probability p_v of it being a fraudulent entity. Then, we train the model using the weighted cross-entropy loss function.

$$p_v = MLP(h_v^{(L)}), \tag{12}$$

$$\mathcal{L} = -\sum_{v \in \mathcal{V}} (\gamma y_v \log(p_v) + (1 - y_v) \log(1 - p_v)), \tag{13}$$

where label $y_v = 1$ denotes fraud while $y_v = 0$ denotes benign, γ is the ratio of fraud labels to benign labels.

4. EXPERIMENTS

4.1. Experimental Setup

4.1.1. Datasets

We investigate the effectiveness of F^{2}GNN on two real-world fraud detection datasets: YelpChi [17] and Amazon [18]. The nodes in the graph of the YelpChi dataset are reviews with 32-dimensional features that have three relations: R-U-R, R-T-R, and R-S-R. The nodes in the graph of the Amazon dataset are users with 20-dimensional features that have three relations: U-P-U, U-S-U, and U-V-U. Table 1 provides the statistics of these two datasets, and detailed descriptions can be found in [4].

Table 1: Dataset and graph statistics.

Dataset	#Nodes (Fraud%)	Relations	#Relations	Avg. Feature Similarity
YelpChi	45,954 (14.5%)	R-U-R	49,315	0.83
		R-T-R	573,616	0.79
		R-S-R	3,402,743	0.77
Amazon	11,944 (9.5%)	U-P-U	175,608	0.61
		U-S-U	3,566,479	0.64
		U-V-U	1,036,737	0.71

4.1.2. Baselines

We compare F^{2}GNN with various GNN approaches. We select GCN [14], GAT [19], and GraphSAGE [20] as general GNN models. We select CARE-GNN [4], PC-GNN [6], AO-GNN [21], H^2-FDetector [8], NGS [9] as state-of-the-art GNN-based fraud detection methods. We select BWGNN [13] and GHRN [22] as state-of-the-art GNN-based anomaly detection methods. Variant F^{2}GNN$_{\setminus A}$ removes the high-pass filter component from F^{2}GNN, and F^{2}GNN$_{\setminus S}$ removes the feature segmentation component.

4.1.3. Experimental Settings

For the F^{2}GNN model, we use Adam as the optimizer with a weight decay of 0.00005 and a learning rate of 0.01. The value of ε is set to 0.1. For the YelpChi dataset, the segmentation dimension $seg\_dim = 8$, the number of segments $seg\_num = 8$, the number of layers $layer\_num = 2$, and $drop\_Message = 0.15$. For the Amazon dataset, the $seg\_dim = 32$, $seg\_num = 2$, $layer\_num = 1$ and $drop\_Message = 0.02$. We set the number of epochs for all models to 500. For others state-of-the-art GNN-based methods, we adopt the parameters as specified by the authors. The dataset is divided in a manner similar to [13].

4.1.4. Evaluation Metrics and Implementation

Considering the class imbalance in fraud detection, we select F1-macro, AUC (Area Under the ROC Curve), and GMean as the evaluation metrics. For general GNN models, we implement them based on DGL [23]. Other baselines were implemented based on the source code provided by the authors. We conduct 10 runs of the experiments and report the average score and standard deviation for all models.

4.2. Results and Analysis

The experimental results are presented in Table 2. Our approach outperforms the state-of-the-art models, demonstrating the effectiveness of F^{2}GNN. The obtained results yield several noteworthy observations that shed light on the following aspects.

Firstly, the general GNN methods and baseline models that treat all edges as a single relation exhibit poor performance on every dataset, while the baseline models based on multi-relation graphs achieve better results. This indicates that each relation contains rich semantic information. F^{2}GNN treating each relation independently ensures the integrity of information.

Then, CARE-GNN and PC-GNN selectively aggregate neighbor features and overcome shortcomings of low-pass filters, but they neglect high-frequency information. H^2-FDetector utilizes node feature similarity to identify heterophily connections. However, randomly dropping features and transforming them at each layer can lead to reduced performance. NGS and BWGNN cannot fully explore the small-portion fraudulent information due to their global modeling approach to feature information. GHRN randomly prunes heterophily edges, but experimental results demonstrate that this approach does not consistently improve performance.

Finally, our method combines feature segmentation and adaptive filtering, and it greatly preserves the integrity of features to mine more fraud information. F^{2}GNN overcomes the limitations of existing approaches.

4.3. Ablation Study

We identify the two key parts of F^{2}GNN, i.e., adaptive filter and feature segmentation, and verify their effectiveness by removing each part, respectively. In Table 2, variant F^{2}GNN$_{\setminus A}$ gets the lowest scores in both datasets. This indicates that high-pass filter can effectively capture high-frequency fraud information in graph-based fraud detection. For variant F^{2}GNN$_{\setminus S}$, it shows significantly poorer performance compared to F^{2}GNN on the YelpChi dataset. On the Amazon dataset, it achieves slightly lower scores compared to except for the F1-macro score, which is easily influenced by the classification threshold.

In Table 1, the YelpChi graph has more fraudsters, but the similarity of node features is high, indicating that fraud features are more

Table 2: Performance comparison on YelpChi and Amazon for opinion fraud detection

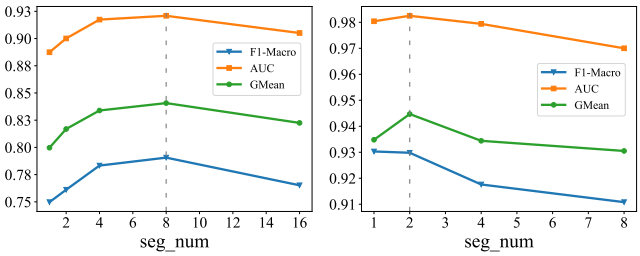
Method	Dataset	YelpChi			Amazon		
	Metric	F1-macro	AUC	GMean	F1-macro	AUC	GMean
Baselines	GCN	0.4979±0.0008	0.5611±0.0005	0.5141±0.0017	0.6625±0.0011	0.8173±0.0009	0.6801±0.0022
	GAT	0.5204±0.0059	0.5703±0.0023	0.5121±0.0184	0.6790±0.0025	0.8308±0.0035	0.6812±0.0039
	GraphSAGE	0.5781±0.0239	0.7409±0.0034	0.6815±0.0049	0.8383±0.0109	0.9149±0.0077	0.8518±0.0077
	CARE-GNN	0.6281±0.0137	0.7918±0.0002	0.7279±0.0035	0.8765±0.0011	0.9425±0.0156	0.8982±0.0015
	PC-GNN	0.6240±0.0665	0.8500±0.0147	0.7543±0.0322	0.8820±0.0053	0.9664±0.0054	0.9095±0.0056
	AO-GNN	0.7042±0.0051	0.8805±0.0008	0.8134±0.0232	0.8921±0.0045	0.9640±0.0020	0.9096±0.0105
	H^2-FDetector	0.7301±0.0014	0.8999±0.0041	0.8232±0.0047	0.8480±0.0433	0.9391±0.0235	0.9108±0.0179
	NGS	0.7754±0.0048	0.9134±0.0021	0.8244±0.0069	0.9202±0.0020	0.9720±0.0043	0.9213±0.0065
	BWGNN(Homo)	0.7311±0.0032	0.8513±0.0069	0.7677±0.0075	0.9181±0.0057	0.9745±0.0035	0.9266±0.0038
	BWGNN(Hetero)	0.7895±0.0044	0.9130±0.0048	0.8299±0.0079	0.9159±0.0061	0.9764±0.0024	0.9213±0.0072
	BHomo-GHRN	0.7312±0.0062	0.8599±0.0063	0.7747±0.0071	0.9199±0.0058	0.9643±0.0090	0.9112±0.0120
	BHetero-GHRN	0.7751±0.0092	0.9077±0.0053	0.8282±0.0069	0.9151±0.0105	0.9706±0.0040	0.9188±0.0079
Ablation	$F^2GNN_{\setminus A}$	0.7248±0.0029	0.8634±0.0037	0.7687±0.0190	0.8843±0.0072	0.9709±0.0007	0.9105±0.0045
	$F^2GNN_{\setminus S}$	0.7497±0.0251	0.8874±0.0134	0.7997±0.0230	**0.9303±0.0051**	0.9814±0.0016	0.9359±0.0059
Ours	F^2GNN	**0.7907±0.0051**	**0.9206±0.0037**	**0.8317±0.0086**	0.9278±0.0042	**0.9825±0.0011**	**0.9447±0.0029**

concealed. In contrast, the Amazon graph exhibits more obvious fraud features. This is why feature segmentation significantly improves performance on the YelpChi graph compared to the Amazon.

4.4. Sensitivity Analysis

In Figure 2, we further evaluate the sensitivity of F^2GNN with respect to the number of segments into which the features are divided. On the YelpChi dataset, the best performance is achieved when the features are divided into 8 segments. On the Amazon dataset, the best performance is obtained with 2 segments.

This is similar to the findings of the ablation study. The YelpChi graph has more concealed fraud features, so it requires more segments to increase the granularity of filtering. In contrast, the fraudulent features are more pronounced in the Amazon dataset. In addition, appropriately increasing the number of segments can enhance filtering performance. Excessive segmentation may introduce unnecessary complexity and lead to a decrease in model performance.



(a) YelpChi (b) Amazon

Fig. 2: Sensitivity analysis with respect to the number of segments into which the features are divided.

5. RELATED WORK

Graph-based Fraud Detection. Graph-based fraud detection methods have gained increasing attention due to their excellent performance. SemiGNN [24] proposes a hierarchical attention mechanism to detect fraudsters. GraphConsis [5] and CARE-GNN [4] address the issue of fraud camouflage by removing dissimilar neighbors before aggregation. PC-GNN [6] and AO-GNN [21] tackle the class-imbalance issue by node resampling and edge pruning, respectively. FRAUDRE [25] integrates four modules into a GNN to address the challenges of graph inconsistency and imbalance. BLS [26] learns to select valuable nodes via the meta gradient of the loss on an unbiased clean validation set. H^2-FDetector [8] propagates different neighbor information by identifying homophilic and heterophilic connections. NGS [9] utilizes a meta-graph search strategy to address fraud detection while maintaining interpretability. BWGNN [13] utilizes a bandpass filter in the spectral domain to capture anomalous information. GHRN [22] devise a label (prediction)-aware edge indicator to prune possibly heterophily edges. Different from that, we segment the user features and apply adaptive filtering to each segment to capture fraudulent information.

6. CONCLUSION

In this paper, we study the adversarial camouflage of fraudulent features and class-imbalance issues in fraud detection. We propose F^2GNN to address these challenges. The approach consists of two key parts: feature segmentation and adaptive filter. They are used respectively to address fraud camouflage and class imbalance issue. Experimental results on two real-world datasets demonstrate that F^2GNN outperforms state-of-the-art methods in fraud detection.

7. ACKNOWLEDGMENTS

The research work is supported by National Key R&D Plan No. 2022YFC3303302, the National Natural Science Foundation of China under Grant (No. 61976204), and the CAAI Huawei MindSpore Open Fund. Xiang Ao is also supported by the Project of Youth Innovation Promotion Association CAS and the Beijing Nova Program. Yang Liu is also supported by the China Postdoctoral Science Foundation (No. 2023M743567).

8. REFERENCES

[1] Can Liu, Qiwei Zhong, Xiang Ao, Li Sun, Wangli Lin, Jinghua Feng, Qing He, and Jiayu Tang, "Fraud transactions detection via behavior tree with local intention calibration," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3035–3043.

[2] Xiaoqian Zhu, Xiang Ao, Zidi Qin, Yanpeng Chang, Yang Liu, Qing He, and Jianping Li, "Intelligent financial fraud detection practices in post-pandemic era," *The Innovation*, vol. 2, no. 4, pp. 100176, 2021.

[3] Wangli Lin, Li Sun, Qiwei Zhong, Can Liu, Jinghua Feng, Xiang Ao, and Hao Yang, "Online credit payment fraud detection via structure-aware hierarchical recurrent neural network.," in *IJCAI*, 2021, pp. 3670–3676.

[4] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 315–324.

[5] Zhiwei Liu, Yingtong Dou, Philip S Yu, Yutong Deng, and Hao Peng, "Alleviating the inconsistency problem of applying graph neural network to fraud detection," in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 1569–1572.

[6] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He, "Pick and choose: a gnn-based imbalanced learning approach for fraud detection," in *Proceedings of the Web Conference 2021*, 2021, pp. 3168–3177.

[7] Qiwei Zhong, Yang Liu, Xiang Ao, Binbin Hu, Jinghua Feng, Jiayu Tang, and Qing He, "Financial defaulter detection on online credit payment via multi-view attributed heterogeneous information network," in *Proceedings of The Web Conference 2020*, 2020, pp. 785–795.

[8] Fengzhao Shi, Yanan Cao, Yanmin Shang, Yuchen Zhou, Chuan Zhou, and Jia Wu, "H2-fdetector: a gnn-based fraud detector with homophilic and heterophilic connections," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1486–1494.

[9] Zidi Qin, Yang Liu, Qing He, and Xiang Ao, "Explainable graph-based fraud detection via neural meta-graph search," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4414–4418.

[10] Lingfei Ren, Ruimin Hu, Yang Liu, Dengshi Li, Junhang Wu, Yilong Zang, and Wenyi Hu, "Improving fraud detection via imbalanced graph structure learning," *Machine Learning*, pp. 1–22, 2023.

[11] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger, "Simplifying graph convolutional networks," in *International conference on machine learning*. PMLR, 2019, pp. 6861–6871.

[12] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen, "Beyond low-frequency information in graph convolutional networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 3950–3957.

[13] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li, "Rethinking graph neural networks for anomaly detection," in *International Conference on Machine Learning*. PMLR, 2022, pp. 21076–21089.

[14] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *in ICLR*, 2017.

[15] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[16] Taoran Fang, Zhiqing Xiao, Chunping Wang, Jiarong Xu, Xuan Yang, and Yang Yang, "Dropmessage: Unifying random dropping for graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023, vol. 37, pp. 4267–4275.

[17] Julian John McAuley and Jure Leskovec, "From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 897–908.

[18] Shebuti Rayana and Leman Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, 2015, pp. 985–994.

[19] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio, "Graph attention networks," *in ICLR*, 2018.

[20] Will Hamilton, Zhitao Ying, and Jure Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[21] Mengda Huang, Yang Liu, Xiang Ao, Kuan Li, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He, "Auc-oriented graph neural network for fraud detection," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1311–1321.

[22] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang, "Addressing heterophily in graph anomaly detection: A perspective of graph spectrum," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 1528–1538.

[23] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, et al., "Deep graph library: A graph-centric, highly-performant package for graph neural networks," *arXiv preprint arXiv:1909.01315*, 2019.

[24] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi, "A semi-supervised graph attentive network for financial fraud detection," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 598–607.

[25] Ge Zhang, Jia Wu, Jian Yang, Amin Beheshti, Shan Xue, Chuan Zhou, and Quan Z Sheng, "Fraudre: Fraud detection dual-resistant to graph inconsistency and imbalance," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 867–876.

[26] Linfeng Dong, Yang Liu, Xiang Ao, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He, "Bi-level selection via meta gradient for graph-based fraud detection," in *Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, April 11–14, 2022, Proceedings, Part I*. Springer, 2022, pp. 387–394.