DualFraud: Dual-Target Fraud Detection and Explanation in Supply Chain Finance Across Heterogeneous Graphs

Bin Wu[1], Kuo-Ming Chao[2], and Yinsheng Li[1(✉)]

[1] School of Computer Science, Fudan University, Shanghai, China
{bwu18,liys}@fudan.edu.cn
[2] Department of Computing and Informatics, Bournemouth University, Bournemouth, UK
kchao@bournemouth.ac.uk

Abstract. In supply chain finance, detecting fraudulent enterprises and transactions is crucial to minimize financial loss. Enterprises and transactions have heterogeneous information and fraud labels, thus, leveraging such information well can simultaneously improve fraud detection performance in enterprise and transaction domains. This paper describes our newly proposed multitask learning framework, DualFraud, which detects fraudulent enterprises and transactions with explainability based on heterogeneous graphs in supply chain finance. The main contributions of this work are the proposed framework that can facilitate these two domains to share and enhance learning and modelling capabilities to improve fraud projection. The explainer component is attached to generate rich and meaningful explanations for risk controllers across enterprise and transaction graphs. Experiments on datasets prove the effectiveness of fraud detection and explainability in both enterprises and transactions. The proposed DualFraud outperforms the other methods in the selected criteria.

Keywords: Fraud Detection · Heterogeneous Graph · Graph Neural Network · Multitask · Explainability · Supply Chain

1 Introduction

Supply chain finance builds upon the trust and credit among suppliers, buyers and financiers to reduce financing costs and manage their cash flow effectively. Fraud has a significant negative impact on supply chain participants and financial institutions, but it is not easy to detect. Fraudsters tend to hide them and prevent them from being uncovered. In recent years, graph neural networks have been gaining popularity in financial applications due to their ability to model complex finance networks and capture individual and structural information [13].

Figure 1 illustrates an example of supply chain finance. The fraud enterprise connects to other fraud enterprises through investment relationship and other

X. Wang et al. (Eds.): DASFAA 2023, LNCS 13946, pp. 370–379, 2023.
https://doi.org/10.1007/978-3-031-30678-5_28

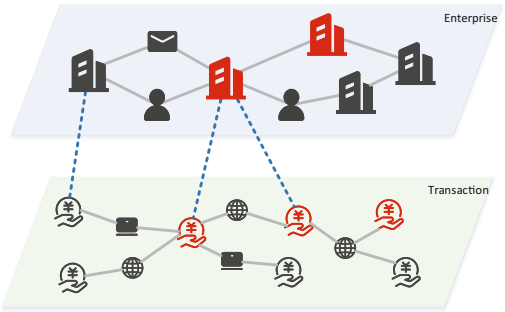


Fig. 1. An example of supply chain finance.

normal enterprises through common persons or emails. The fraud enterprise has conducted several fraud transactions. Transactions are connected through common networks and devices. In this paper, we propose to address three challenges. Firstly, enterprises and transactions have different features and relations. Different graphs need to be designed to capture their characteristics. Besides, business transactions are more dynamic than enterprise records. Decoupling of graphs facilitates the scalability of the subsequent model. Secondly, we need to improve the performance of fraudulent enterprise and transaction detection in the meantime while keeping robustness and flexibility. Fraud detection of enterprises and transactions are two different but related tasks. It is necessary to devise a novel model to improve the performance of both tasks simultaneously by leveraging data richness and diversity in both domains. Also, we need to provide explanations of fraud detection across graphs to facilitate further processes in the business unit. In our multi-graph setting, the model needs to provide explanations from both enterprise and transaction perspectives.

To tackle the above issues, we propose DualFraud, a dual-target fraud detection and explanation framework for enterprises and transactions. DualFraud can effectively predict the legitimacy of enterprises and transactions, and generate explanations on multiple heterogeneous graphs in the multitask setting. We construct two separate heterogeneous graphs of different node types to tackle multi-source and heterogeneous information. We use metapath [11] aggregated GCNs [3] to generate embeddings of enterprises and transactions separately. To enhance both tasks, we consider fraud enterprise and transaction detection together from multitasking learning. We use Att-BLSTM [18] to learn representations of the transaction history. We extend GNNExplainer [15] to multiple graphs by utilizing the attention scores. The contributions of this work are as follows:

1. We present a novel multitask learning framework DualFraud to capture and represent both fraud enterprises and transactions in a unified way.
2. We add explainability into DualFraud to form an end-to-end fraud detection mechanism in supply chain finance.
3. We have conducted experiments on datasets to show its superiority compared with the state-of-the-art models in fraud detection.

2 Literature

For fraud transaction detection, IHGAT [5] constructs a heterogeneous transaction-intention network in e-commerce platforms to leverage the cross-interaction information over transactions and intentions. xFraud [10] constructs a heterogeneous graph to learn expressive representations. For enterprises, ST-GNN [14] addresses the data deficiency problem of financial risk analysis for SMEs by using link prediction and predicts loan default based on a supply chain graph. HAT [17] proposes a heterogeneous-attention-network-based model to facilitate SME bankruptcy prediction. These methods focus on a single task of fraud detection only. For GNN-based fraud detection in the multitask setting, MvMoE [4] proposes a multi-view and multitask learning-based approach to solve credit risk and limit forecasting simultaneously. GraphRfi [16] proposes a framework with GCN and neural random forest to solve robust recommendation and fraudster detection. These methods cannot be used directly for dual-target fraud detection tasks. Also, these methods use only one GNN component, and information cannot be shared and propagated from different perspectives. For explainability, GNNExplainer [15] is an model-agnostic approach for providing interpretable explanations for predictions. xFraud [10] extends it to heterogeneous graphs. Our proposed framework takes this work further to multiple heterogeneous graphs.

3 Methodology

The DualFraud framework is shown in Fig. 2, which consists of two key modules: detector and explainer.

3.1 Detector of DualFraud

The Detector is responsible for learning from labeled data and detecting possible frauds. To fully utilize the labels and features, two heterogeneous graphs of different node types are constructed: a heterogeneous enterprise graph $\mathcal{G}_e = \{\mathcal{V}, E\}$ and a heterogeneous transaction graph $\mathcal{G}_t = \{\mathcal{U}, E'\}$. We adopt metapath and GCN to capture the information of nodes and heterogeneous topological neighborhood structure simultaneously. We assume that initial node representations of each enterprise node $v \in \mathcal{V}$ is h_v^0. The kth layer embedding is:

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_k \sum_{u \in N(v) \cup v} \frac{\mathbf{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}}\right) \tag{1}$$

where $\sigma(\cdot)$ is the activation function, and $\mathbf{W}_k$ is the weight matrix. $N(v)$ is the set of nodes in the neighborhood of node v.

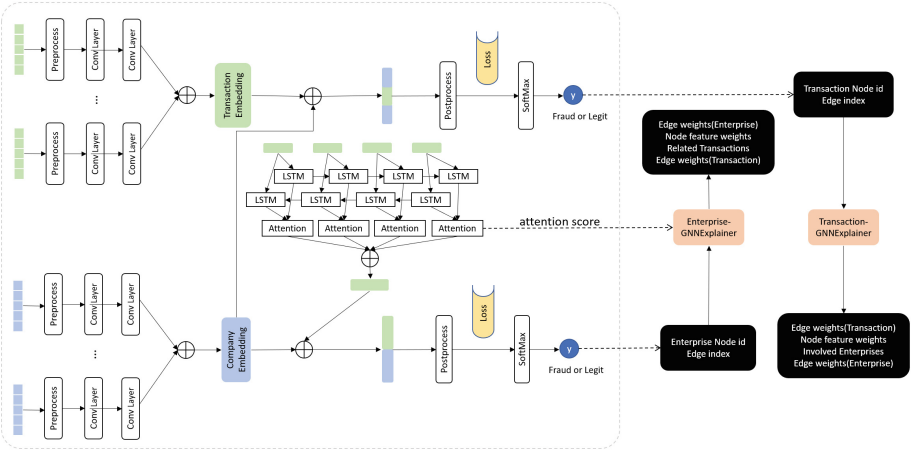


Fig. 2. DualFraud detector and explainer.

We apply two graph convolutional layers, with skip connections, to the node representation. The final representation is $\mathbf{z}_v = \mathbf{h}_v^K$. We select several metapaths manually based on domain experts to capture different relations. We concatenate representations from different metapaths as the embedding of the enterprise node, $\mathbf{z}_v^F = \bigoplus_{i=1}^{M} \mathbf{z}_v^i$, where $\oplus$ denotes the concatenation of two vectors, and M is the number of metapaths. Similarly, we get the embedding $\mathbf{z}_u^F$ of transaction nodes from the heterogeneous transaction graph $\mathcal{G}_t = \{\mathcal{U}, \mathcal{E}'\}$.

We use bidirectional LSTM and Attention to learn the representation of transaction history and concatenate it with node embedding $\mathbf{Z}_v^F$ of the enterprise. The node embeddings of transactions are $\mathbf{Z}_{uT}^F = \{z_{u1}^F, z_{u2}^F, ..., z_{ut}^F\}$ from the transaction graph. For each transaction, the LSTM layer computes the hidden state h_t at time t. The BLSTM contains two LSTM layers in opposite directions. The output of the i^{th} transaction is the element-wise sum of outputs from forward and backward LSTM layers: $h_i = \left[\overrightarrow{h_i} + \overleftarrow{h_i}\right]$. We adopt the attention mechanism to capture transactions that are important to the enterprise. The h_i is fed into a one-layer MLP to get the hidden representation u_i. We can get the importance weight α_i through a Softmax function.

$$\begin{aligned} u_i &= \tanh\left(W_a h_i + b_a\right), \\ \alpha_i &= \frac{\exp\left(u_i\right)}{\sum_i \exp\left(u_i\right)}, \\ \mathbf{z}_v^A &= \sum_i \alpha_i h_i \end{aligned} \tag{2}$$

The concatenated embeddings of enterprises are defined as: $\mathbf{z}_v^{new} = \mathbf{z}_v^F \oplus \mathbf{z}_v^A$. The attention scores of historical transactions of enterprise v are defined as $\mathbf{Att}_v = \{\alpha_1, \alpha_2, ..., \alpha_n\}$, where n is the number of transactions of enterprise v.

For transactions, we concatenate embeddings of the sender $\mathbf{z}_{vs}^{F}$ and receiver $\mathbf{z}_{vr}^{F}$ to get the embeddings: $\mathbf{z}_{u}^{new} = \mathbf{z}_{u}^{F} \oplus \mathbf{z}_{vs}^{F} \oplus \mathbf{z}_{vr}^{F}$.

We use the standard supervised cross-entropy loss to train the model. Instead of training these two tasks separately, we combine their losses and jointly minimize the following loss function:

$$\mathcal{L} = \lambda \mathcal{L}_{enterprise} + (1 - \lambda)\mathcal{L}_{transaction} \tag{3}$$

where λ is a hyper-parameter to balance the effect of both parts.

3.2 Explainer of DualFraud

We add explainability to DualFraud by extending GNNExplainer [15] to multiple heterogeneous graphs. The model's prediction of node v is given by $\hat{y} = \Phi\left(G_c(v), X_c(v), I_c(v)\right)$, where Φ is the model trained in the detection process. The prediction is determined by graph structural information $G_c(v)$, node feature information $X_c(v)$, and information from the other graph $I_c(v)$. It implies that we need to consider these three aspects to explain $\hat{y}$. Formally, the explainer component generates explanations for the prediction

$$\hat{y} = \left(G_{vS}, X_{vS}^{F}, \underset{i \in k}{\cup} G_{uSi}, \underset{i \in k}{\cup} X_{uSi}^{F}\right) \tag{4}$$

where G_{vS} is a small subgraph of the target node, G_{uS} are subgraphs from the other graph of related nodes. X_{vS} and X_{uS} are the associated feature of G_{vS} and G_{uS}. X_{vS}^{F} and X_{uS}^{F} are small subsets of node features (masked out by the mask F, i.e., $X_{vS}^{F} = \left\{x_j^F \mid v_j \in G_{vS}\right\}, X_{uS}^{F} = \left\{x_j^F \mid v_j \in G_{uS}\right\}$).

For enterprises, we choose the top $\mathbf{k}$ transactions of attention scores $\mathbf{Att}_v$ generated by the Att-BLSTM component since attention reflects the importance of this transaction to the enterprise. For transactions, we apply the explainer to the involved enterprises. The subgraphs of enterprises and transactions are linked together as the final explanation result.

4 Experiment

4.1 Datasets and Comparing Methods

This research adopts two different datasets, one synthetic and one real-world. The statistical information describing datasets is shown in Table 1.

Synthetic Dataset. We construct synthetic datasets referring to the design of experimental datasets of GNNExplainer [15]. For enterprise relationships, we construct a Tree-Cycles dataset. It starts with a base 9-level balanced binary tree and 20 six-node cycle motifs, which are attached to random nodes of the base graph. The resulting graph is further perturbed by adding 0.1N random edges. The tree-like structure is similar to the relationships of enterprises in supply

Table 1. Statistical information of datasets.

	Nodes (Fraud%)	#Features	Relation	#Edges
Synthetic Enterprise	1143 (10.5%)	8	C-C	1181
Synthetic Transaction	4575 (10.5%)	8	T-T	4749
Real-world Enterprise	13489 (26.4%)	89	C-I-C C-C C-M-C ALL	53874 15908 139413 209195
Real-world Transaction	50000 (1.2%)	23	T-A-T	206666

chain. Nodes in the base graph are labeled with 0 to represent normal enterprises. The structure of cycle motifs is similar to the relationship in self-financing fraud, one of the common types of supply chain accounts-receivable frauds[1]. Nodes in the cycle motifs are labeled with 1 to represent fraud enterprises. Nodes in the same class have normally distributed feature vectors. Similarly, for transaction relationships, we construct a base 11-level balanced binary tree and 80 six-node cycle motifs. Every fraud transaction is mapped to two random fraud enterprises, while normal transactions are mapped to random enterprises.

Real-World Dataset. We use public real-world datasets of HAT [17] and BankSim [8] to construct our experimental dataset. The SME's Bankruptcy Dataset of HAT contains the board member network and shareholder network for 13489 companies, in which 3563(26.4%) companies go bankrupt. Specifically, 1000 companies are selected firstly, which located in a south-eastern city in China and went bankrupt in 2018. Then, all the shareholders and board members for these enterprises are collected and this process is repeated for the collected enterprises twice. BankSim is based on a sample of aggregated transactional data provided by a bank. Every transaction in the BankSim dataset has the label of fraud(1.2%) or normal(98.8%). We define entities in BankSim with at least one fraudulent transaction as fraudulent and match them with fraudulent enterprises in the HAT data. The rest of the entities are matched with normal enterprises.

We evaluate the performance by comparing with the following methods:

- GraphSage [2]: A popular inductive GNN framework generates embeddings by sampling and aggregating features from a node's local neighborhood.
- GEM [7]: A heterogeneous GNN approach for detecting malicious accounts which adopts attention to learn the importance of different types of nodes.
- SemiGNN [12]: A semi-supervised graph attentive network for financial fraud detection that utilizes the multi-view labeled and unlabeled data.
- GraphConsis [6]: A model that tackles inconsistency problems in applying GNNs in fraud detection problems.
- RioGNN [9]: A SOTA fraud detection model in reinforced, recursive and flexible neighborhood selection guided multi-relational GNN architecture.
- DualFraud-S: Version with DualFraud's embedding sharing module removed.

Due to the class imbalance in fraud detection, we select Macro-F1 and AUC to evaluate the performance of the models.

[1] http://www.nifd.cn/ResearchComment/Details/2582.

4.2 Experimental Settings and Implementation

In DualFraud, we set the hidden embedding size to 16, the number of graph convolution layers to 2, the number of recurrent layers to 1, and the dimension of attention layer to 5. For model optimization, we use Adam optimizer and set the learning rate to 0.0003. The codes for GraphSage, GEM, SemiGNN, and GraphConsis are from DGFraud-TF2 [1]. The codes for other models come from respective authors' implementations. For GraphSage which adopts homogeneous graphs, the edges of different types are treated as the same. For the datasets, we distribute them according to the ratio 68:12:20 for training, validation, and test set, respectively. The source code of our model and constructed datasets are available[2]. All experiments are conducted by Python 3.6, NVIDIA GeForce GTX 1080 and AMD Ryzen 5 2600.

4.3 Results

Fraud Detection. We present experimental results in Table 2 of the fraud enterprise and transaction detection tasks on the synthetic and real-world datasets. In both synthetic and real-world datasets, DualFraud outperforms all the compared methods in F1 and AUC. In the real-world dataset, SemiGNN cannot complete the experiment because of out-of-memory. For fraud transaction detection in real-world dataset, multiple methods including GraphSage and GEM cannot perform discrimination effectively due to the severe class imbalance problem. They predict all transactions in the test dataset as normal. While DualFraud-S performs poorly compared with other methods, DualFraud improves the performance by a large margin. It demonstrates the effectiveness of sharing embeddings to exploit both enterprise and transaction information. By sharing information through the multitask framework, information from other networks is used as side information to enhance the features of the current task. This increases the difference of features of samples from different categories, which is more effective in real-world transaction dataset that has severe class imbalance. On the task of fraud enterprise detection in real-world dataset, DualFraud is only marginally better than the baselines compared with other tasks. The results are consistent with the experimental results in the paper [17]. This may be due to the fact that the features and relationships built by the dataset are not effective in classification task.

Parameter Study. For the multitask setting, we study the parameter sensitivity of weights of different tasks in Fig. 3. We run experiments on real-world dataset using different λ parameters in Eq. 3. We can notice that the larger the lambda value,

[2] https://github.com/anonymousDualFraud/DualFraud.

Table 2. Comparison of methods on Synthetic(S.) dataset and Real-world(R.) dataset.

	S. Enterprise		S. Transaction		R. Enterprise		R. Transaction	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC
GraphSage	0.6790	0.6250	0.6286	0.6088	0.6465	0.6303	0.4971	0.5000
GEM	0.9069	0.9069	0.8613	0.8947	0.6436	0.6354	0.4971	0.5000
SemiGNN	0.6216	0.7985	0.4281	0.6137	OOM	OOM	OOM	OOM
GraphConsis	0.5398	0.5343	0.5102	0.5150	0.6498	0.6338	0.5107	0.5074
RioGNN	0.6177	0.6524	0.5566	0.5667	0.6659	0.6612	0.5509	0.5707
DualFraud-S	0.8743	0.8468	0.8958	0.8867	0.4595	0.5122	0.8841	0.8581
DualFraud	**0.9758**	**0.9583**	**0.9397**	**0.9458**	**0.6974**	**0.7492**	**0.8891**	**0.8792**

i.e., the higher the weight of the task, the smaller the loss value in task 1. Since the method performs much better on task 2 than on task 1, different lamda parameters have no significant effect on task 2. Thus, the weights can be adjusted according to the performance of different tasks in practical use.

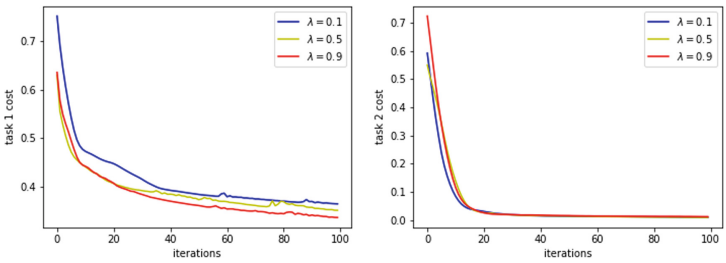


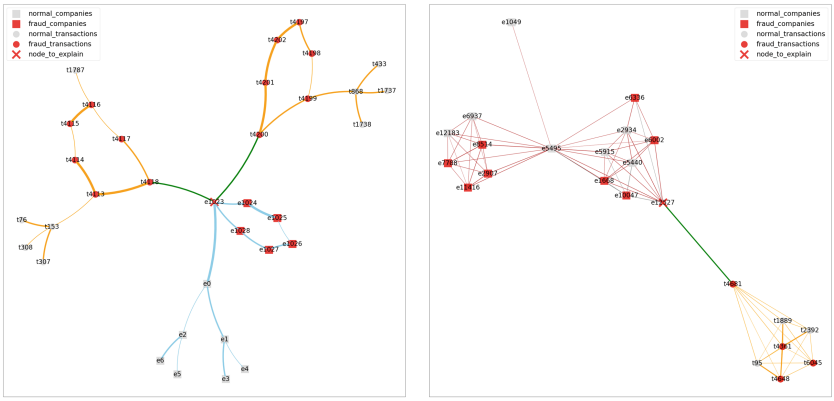
Fig. 3. Training loss in different λ parameter.

Explainability. We present case studies on flagging frauds in Fig. 4. Different shapes and colored edges indicate different types of nodes and relations respectively. The edges' thickness represents the relations' importance in the prediction. The thicker the more important. We use ground truth labels as the color of nodes of enterprises and transactions. If the edge between the red node and the target node is bolded, it indicates that the explainer provides a valid explanation. If all the provided transaction nodes connected to the target enterprise node are red, it indicates that the explainer found the suspicious transactions accurately.

In Fig. 4(a), we present an explanation result for a fraud enterprise(node *e1023*) in the synthetic dataset. The explainer catches most of the critical edges with other fraud enterprises in the Cycle motif of the target node. Also, the explainer identifies the enterprise's fraud transactions(node *t4118* and node *t4200*) with the highest attention score. Then the explainer shows the most

suspicious edges from the perspective of transactions. The subgraph provides good explanations from enterprise and transaction perspectives.

In Fig. 4(b), we present an explanation result for a fraud enterprise(node *e12527*) in the real-world dataset. The enterprise *e12527* has the same shareholders as enterprises *e6336* and *e10047*, both of which are fraudulent enterprises. The enterprise *e12527* has the same board members as enterprises *e2934*, *e5440*, *e1668*, *e5915* and *e5495*, in which only enterprise *e1668* is fraudulent. Although enterprise *e5495* is normal, it has the same board members as several fraudulent enterprises including *e2907*, *e7788*, *e11416* and *e8514*. The transaction(node *t4681*) with the highest attention score in the history of transactions of enterprise *e12527* is a fraud. This transaction is related to multiple fraudulent transactions. This demonstrates the ability of DualFraud to identify suspicious enterprises and transactions, even when the proportion of normal and fraud in the neighborhood is similar.



(a) Explanation in the synthetic dataset

(b) Explanation in the real-world dataset

Fig. 4. Explanation of fraud enterprises and transactions.

5 Conclusion

The paper proposes an end-to-end GNN framework called DualFraud in supply chain financial fraud detection. The experimental results demonstrate the effectiveness and practicality of DualFraud on fraud detection and explanation. For future work, when faced with data from more sources, the model needs to handle data and provide explanations from more perspectives. Also, it needs to address conflicts between data from different sources. In transactions, using dynamic graph neural networks to deal with the transaction graph in the model is worthwhile for further investigation.

Acknowledgements. The work has been supported by the National Key Research and Development Program of China (Grant No. 2019YFB1404904).

References

1. 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). https://doi.org/10.1145/3340531.3411903
2. Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NeurIPS, pp. 1024–1034 (2017)
3. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
4. Liang, T., et al.: Credit risk and limits forecasting in e-commerce consumer lending service via multi-view-aware mixture-of-experts nets. In: WSDM, pp. 229–237 (2021)
5. Liu, C., Sun, L., Ao, X., Feng, J., He, Q., Yang, H.: Intention-aware heterogeneous graph attention networks for fraud transactions detection. In: KDD, pp. 3280–3288 (2021). https://doi.org/10.1145/3447548.3467142
6. 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). https://doi.org/10.1145/3397271.3401253
7. Liu, Z., Chen, C., Yang, X., Zhou, J., Li, X., Song, L.: Heterogeneous graph neural networks for malicious account detection. In: CIKM, pp. 2077–2085 (2018). https://doi.org/10.1145/3269206.3272010
8. Lopez-Rojas, E.A., Axelsson, S.: Banksim: A bank payments simulator for fraud detection research. In: EMSS, pp. 144–152 (2014)
9. Peng, H., Zhang, R., Dou, Y., Yang, R., Zhang, J., Yu, P.S.: Reinforced neighborhood selection guided multi-relational graph neural networks. TOIS **40**(4), 1–46 (2021)
10. Rao, S.X., et al.: xFraud: explainable fraud transaction detection. In: Proceedings of VLDB Endow, vol. 15, no. 3, pp. 427–436 (2021). https://doi.org/10.14778/3494124.3494128
11. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: meta path-based top-k similarity search in heterogeneous information networks. Proc. VLDB Endow. **4**(11), 992–1003 (2011)
12. Wang, D., et al.: A semi-supervised graph attentive network for financial fraud detection. In: ICDM, pp. 598–607 (2019). https://doi.org/10.1109/ICDM.2019.00070
13. Wang, J., Zhang, S., Xiao, Y., Song, R.: A review on graph neural network methods in financial applications. arXiv preprint arXiv:2111.15367 (2021)
14. Yang, S., et al.: Financial risk analysis for smes with graph-based supply chain mining. In: IJCAI, pp. 4661–4667 (2020)
15. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnnexplainer: generating explanations for graph neural networks. In: NeurIPS, pp. 9240–9251 (2019)
16. Zhang, S., Yin, H., Chen, T., Nguyen, Q.V.H., Huang, Z., Cui, L.: Gcn-based user representation learning for unifying robust recommendation and fraudster detection. In: SIGIR, pp. 689–698 (2020). https://doi.org/10.1145/3397271.3401165
17. Zheng, Y., Lee, V.C.S., Wu, Z., Pan, S.: Heterogeneous graph attention network for small and medium-sized enterprises bankruptcy prediction. In: KDD, pp. 140–151 (2021). https://doi.org/10.1007/978-3-030-75762-5_12
18. Zhou, P., et al.: Attention-based bidirectional long short-term memory networks for relation classification. In: ACL, pp. 207–212 (2016)