Heterogeneous Graph Neural Networks for Fraud Detection and Explanation in Supply Chain Finance

BinWu[a],Kuo-MingChao[a,b],YinshengLi[a],*

[a]School ofComputerScience, FudanUniversity, Shanghai,China

[b]School ofArts,Humanities, andSocial Sciences,UniversityofRoehampton, London,UK

It is a critical mission for financial service providers to discover fraudulent borrowers in a supply chain. The borrowers' transactions in an ongoing business are inspected to support the providers' decision on whether to lend the money. Considering multiple participants in a supply chain business, the borrowers may use sophisticated tricks to cheat, making fraud detection challenging. In this work, we propose a multitask learning framework, MultiFraud, for complex fraud detection with reasonable explanation. The heterogeneous information from multi-view around the entities is leveraged in the detection framework based on heterogeneous graph neural networks. MultiFraud enables multiple domains to share embeddings and enhance modeling capabilities for fraud detection. The developed explainer provides comprehensive explanations across multiple graphs. Experimental results on five datasets demonstrate the framework's effectiveness in fraud detection and explanation across domains.

Fraud Detection, Supply Chain Finance, Graph Neural Network, Multitask Learning, Graph Explainability, Heterogeneous Graph,

Introduction

In recent years, the supply chain finance market has developed rapidly, effectively alleviating the financing difficulties of many small and medium-sized enterprises (SMEs). However, the supply chain is not just a linear chain composed of upstream and downstream enterprises. The transaction, social, and partnership relationships between enterprises constitute a complex supply chain network. The data in the supply chain consists of commercial flow, information flow, capital flow, and logistics. The multi-source and heterogeneous data and relationships in the supply chain bring challenges to fraud detection in supply chain finance. The length and complexity of transactions create opportunities for fraudsters(Katz 2016), even though they are processed by a set of technological solutions. Many previous works (West and Bhattacharya 2016; Hassan et al. 2020; Albashrawi 2021) have utilized machine learning and deep learning methods to analyze fraud in individual enterprises or transactions. They have achieved limited success, as they pay little attention to studying the massive interactions between enterprises and their business transactions in the supply chain. Graph neural networks(Jianian Wang et al. 2021) have attracted the attention of researchers from various fields due to their powerful ability to learn attributes and relationships. As a natural network, the supply chain can be effectively mined by combining graph neural networks.

In supply chain finance, accounts receivable financing is the most widely used model(Hu et al. 2022). It has become popular in many countries, such as the United States and China(Yan et al. 2021). At the end of July 2023, accounts receivable of industrial enterprises in China amounted to RMB 23.11 trillion, reflecting a year-on-year increase of 9.7%[1]. Supply chain accounts receivable is a financing method in which core enterprises transfer accounts receivable to banks and suppliers to obtain bank loans. Further, a supply chain financial services platform in China makes receivables detachable, holdable, and transferable with the help of blockchain technology. A supply chain accounts receivable can be split and transferred between multiple suppliers to increase flexibility and exposure.

This research intends to address the following three challenges in fraud detection within supply chain finance:

Firstly, effectively exploiting heterogeneous information from multiple views in fraud detection. Multi-view perspectives in supply chain finance contain different levels. Firstly, it contains the perspectives of multiple different entities. Secondly, each entity itself contains multiple different attributes and relationships. In the supply chain, there are heterogeneous types of information concerning a transaction, such as amounts, payment tokens, and payout time. Previous works(C. Liu et al. 2021; S. X. Rao et al. 2021) have constructed transaction graphs for fraud detection to capture interactions between transactions. However, unlike consumer finance, supply chains involve lengthy and complex transactions between enterprises, which contain plenty of information that can be exploited. There are heterogeneous types of information concerning an enterprise, such as registration time, registered capital, actual capital, shareholders, executives, legal persons, telephone numbers, and email addresses. Previous studies(Shuo Yang et al. 2020; Koh et al. 2007; Malhotra, Gosain, and Sawy 2005) have analyzed and proved the strong correlation between supply chain relationships and SMEs' financial risk through exploratory factor analysis. Constructing different graphs to capture various entities from different perspectives enhances the framework's capability and flexibility for representation learning.

Secondly, improving fraud detection performance by utilizing the correlations between multiple views. Fraud labels exist in different views. For example, in enterprises, it includes credit risks, bankruptcy, etc. In transactions, it mainly refers to fraudulent transactions and loan defaults. Fraud labels of different entities are correlated. For example, a company that has already lost credit will likely apply for a loan that will not be repaid. As well, a company with a record of multiple fraudulent transactions within the platform is less reputable. It should also be noted that class imbalance problem is more apparent in transactions(Z. Li et al. 2021; Somasundaram and Reddy 2019). Most previous fraud detection methods(C. Liu

[1] https://www.gov.cn/lianbo/bumen/202308/content_6900436.htm

et al. 2021; S. X. Rao et al. 2021; Shuo Yang et al. 2020; Zheng et al. 2021) focused on only one of these tasks. Developing a model that leverages the richness and diversity of data in both domains is essential to enhance performance.

Finally, providing explanations of fraud predictions across multiple views. Fraud detection is an aid to risk controllers as they need some understanding of the fraud before acting. Explainability is critical to supply chain finance fraud detection. Due to the multiple sources of information in supply chain finance, the explainer needs to provide richer content. While existing works(Ying et al. 2019; Luo et al. 2020; S. X. Rao et al. 2021) offer explainability, they are unsuitable for multiple graphs.

To tackle these issues, we propose MultiFraud, a multitask framework for fraud detection and explanation based on heterogeneous graph neural networks.

In order to address the challenges of handling multi-source, heterogeneous information, we construct heterogeneous graphs for different views separately to maintain their semantics. We utilize heterogeneous GNNs to capture characteristics and heterogeneous relationship properties fully. To construct correlations between views, we propose an attention-based component to share the embedding of entities. We develop an explainer component to generate feature and edge weighs on multiple graphs to provide explainability.

The contributions are summarized as follows:

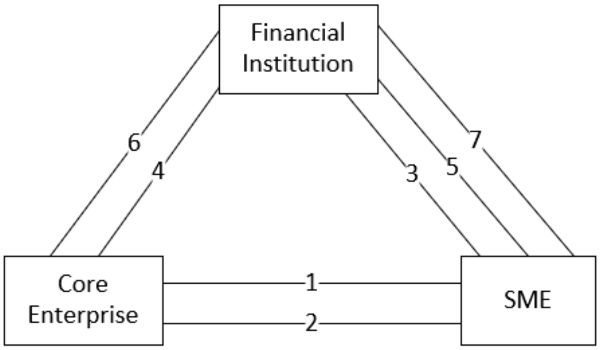
1. To the best of our knowledge, we are the first to tackle fraud detection for multiple views simultaneously in supply chain finance.
2. We propose a multitask learning framework, MultiFraud, with heterogeneous GNNs to detect fraud in supply chain finance.
3. MultiFraud can provide comprehensive explanations on multiple heterogeneous graphs.
4. We conduct experiments on five datasets to evaluate the effectiveness of MultiFraud. The results show that it outperforms the state-of-the-art methods.

The rest of the paper is structured as follows: Section 2 introduces the business model. Section 3 surveys the related works and analyzes their limitations in supply chain fraud detection. Section 4 describes the heterogeneous graph construction and detailed components of the MultiFraud framework. Section 5 reports the experimental evaluation results. Section 6 concludes the work and highlights future research directions.

Accounts Receivable Financing

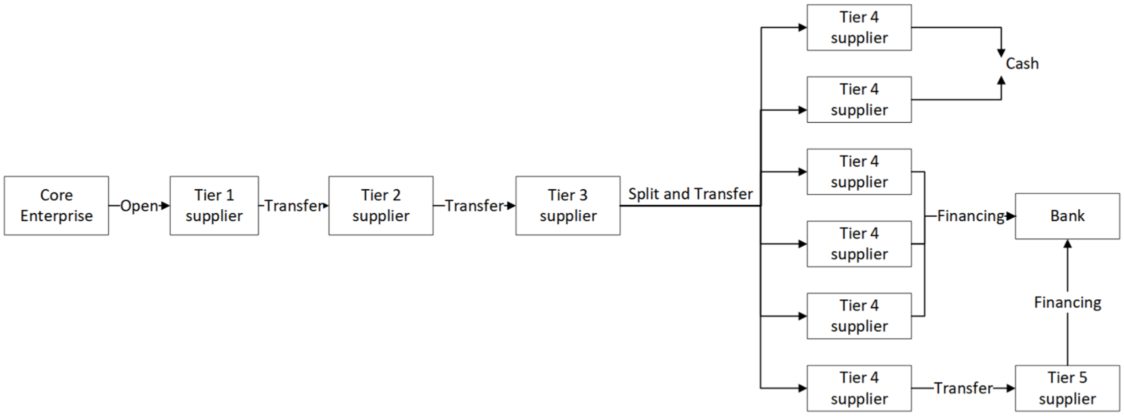
This section describes the detailed business process of supply chain accounts receivable financing. The Property Law of 2007, for the first time at the legal level, clarified the inclusion of accounts receivable as pledges in the scope of security for movable assets in China. Accounts receivable financing is a way for financial institutions to provide financing to SMEs based on actual transactions between core enterprises and upstream and downstream SMEs in the supply chain. This helps alleviate the financing difficulties and capital constraints of SMEs. We present the business model of accounts receivable financing in Fig. 1, which contains three main roles: financial institution, core enterprise and SME. The specific business processes are as follows:

1. The SME signs a purchase contract with the core enterprise.
2. The SME gets the accounts receivable from the core enterprise.
3. The SME issues an application for supply chain accounts receivable financing to the financial institution.
4. The financial institution conducts a pre-load investigation on accounts receivable, documents and credits.
5. The financial institution signs a financial agreement to lend to the SME and informs the core enterprise of the receivable payment.
6. The core enterprise pays accounts receivable back to the financial institution by the due date.
7. The bank transfers the remaining funds to the SME.



Business process of supply chain accounts receivable financing.

Ouyeel, a supply chain financial service platform of China Baowu (one of the world's top 500 companies), has launched digital asset certificates in the form of accounts receivable. The transferable and detachable characteristics of digital assets allow the credit of core companies in the supply chain to extend to more SMEs. We provide an example in Fig. 2. On August 29, a core enterprise (an enterprise in the steel industry) opened a digital asset of RMB 5.97 million to its first-tier supplier (a trading service enterprise). As of September 30, a total of 10 flowthroughs and partially transferred assets, as well as 5 financing operations have occurred for this digital asset. Fig. 2 shows a multi-level flow of accounts receivable with different tiers of suppliers and financing institutes. However, this transferable and detachable model makes the transaction pattern more complex(Chicha et al. 2021), and fraud is more likely to occur and harder to detect.



An example of multi-level flow of accounts receivable.

Literature

[tab:freq]

For risk management systems in supply chain finance, research(Song 2021) studies the establishment of a risk management system. It points out the importance of utilizing big data analysis and machine learning tools. Research(Conforti et al. 2013) proposes a distributed, sensor-based architecture to monitor risks in business processes. Another research(He and Tang 2012) builds a system for visualization of order, logistics and stock in supply chain finance. However, this system only visualizes business information and has not analyzed and visualized supply chain risks. Current risk management platforms(Wu, Liu, and Zhang 2021) mainly rely on expert rules and risk indicator models.

Research(Deng and Yu 2017) summarizes risk assessment methods for the risk management of accounts receivable financing, including principal component analysis, artificial neural networks, analytic hierarchy processes, logistic regression analysis and fuzzy comprehensive evaluation. Another research(Aboutorab et al. 2021) assesses the suitability of the techniques used for risk identification to be effective in the current networked supply chain environment. Previous studies have adopted machine learning techniques for fraud detection in supply chain finance, including distributed CNN(Zhou et al. 2020), Rpart, C5, Random Forest, SVM(Constante-Nicolalde, Guerra-Terán, and Pérez-Medina 2020), and XGBoost(Wan 2021).

Graph neural networks have gained popularity in various domains, including knowledge graphs(Shuang Yang and Cai 2022), recommendations(Xiong et al. 2022), social networks(Ran et al. 2022) and traffic networks(Peng et al. 2020). GNNs have been used in fraud detection, which can be classified into homogeneous and heterogeneous graphs. FdGars(Jianyu Wang et al. 2019), GeniePath(Ziqi Liu et al. 2019), FD-NAG(C. Wang et al. 2021) construct homogeneous graphs and employ GNNs for fraud detection. For heterogeneous graphs, GraphConsis(Zhiwei Liu et al. 2020), CARE-GNN(Dou et al. 2020), PC-GNN(Y. Liu et al. 2021) construct graphs containing one type of node and multiple relations. GAS(A. Li et al. 2019) incorporates a heterogeneous graph and a homogeneous comment graph to generate embeddings. GEM(Ziqi Liu et al. 2018), SemiGNN(D. Wang et al. 2019), BotSpot++(Zhu et al. 2021), MAFI(Jiang et al. 2021) adopt heterogeneous graphs for fraud detection. The benefit of a heterogeneous graph is that it can model multiple-dimensional information and the relationships in one graph for learning and viewing the results. Most previous works focused on consumer finance and spam review, in which users are individuals, not enterprises. They did not consider either complex transactions or complicated enterprise relations in their approaches.

ST-GNN(Shuo Yang et al. 2020) conducts data analyses to reveal the impact of supply chain relationships on the financial risk analysis of SMEs. It proposes a spatial-temporal aware GNN to predict loan default. HAT(Zheng et al. 2021) predicts the bankruptcy of enterprises based on heterogeneous GNN with an attention mechanism. IHGAT(C. Liu et al. 2021) constructs intentions from user behaviors and designs a heterogeneous network, including transactions and intentions. xFraud(S. X. Rao et al. 2021) builds a heterogeneous transaction graph containing buyers, payment tokens, shipping addresses and emails to learn transaction representations. These GNN-based methods are targeted at single views and single tasks.

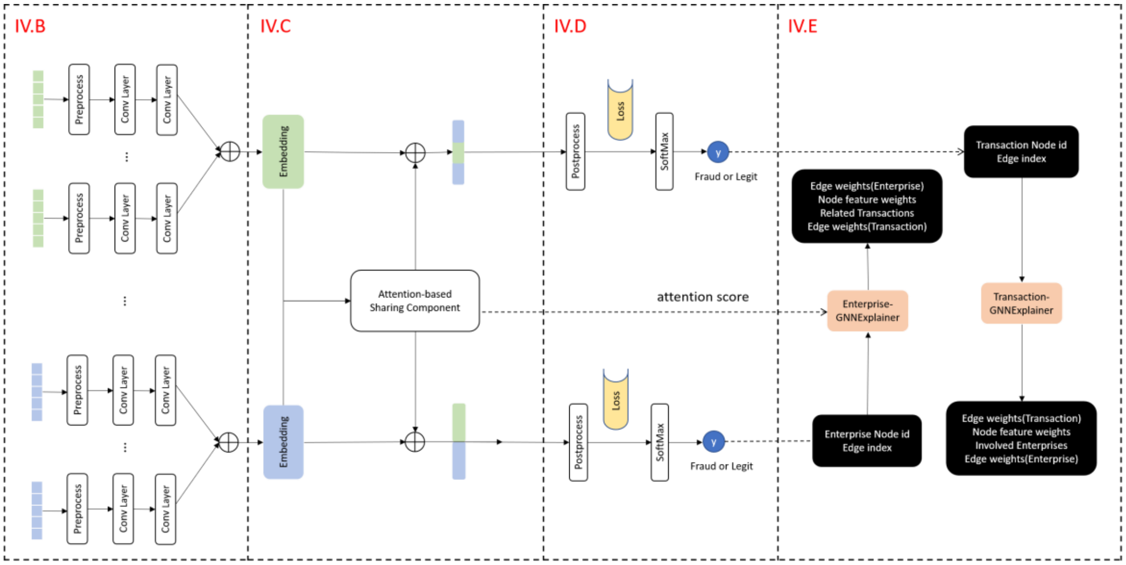
Also, our work is related to fraud detection through a multitask framework. GraphRfi(Zhang et al. 2020) proposes to conduct recommendation and fraud detection by GCN and neural random forest. MvMoE(Liang et al. 2021) proposes solving credit risk and limit forecasting simultaneously by a mixture-of-experts network. It combines heterogeneous multi-view data, including user profiles, sequential behaviors, and social relations. MLP, bidirectional LSTM, and GNN are adopted to encode each view's features. Current multi-task solutions are designed for the same node domain and cannot simultaneously handle tasks for different entities. Besides, they only integrate information from different views at the multi-task learning level. There is no direct interaction between multiple views.

In terms of explainability, demand continues to rise as black-box methods are increasingly being employed(Barredo Arrieta et al. 2020), especially in regulated financial services(Bussmann et al. 2020). Compared with other domains such as image and text, the explainability of GNNs is at an early stage(Yuan, Yu, et al. 2020). The surveys(P. Li et al. 2022; Yuan, Yu, et al. 2020) report several approaches proposed recently and analyze their pros and cons, including GNNExplainer(Ying et al. 2019), PGExplainer(Luo et al. 2020), SubgraphX(Yuan et al. 2021), XGNN(Yuan, Tang, et al. 2020), etc. For explainability in fraud detection, GCAN(Lu and Li 2020) uses co-attention weights derived from the model to generate explainability on evidential words of fake news. Know-GNN(Y. Rao et al. 2021) uses graph functional dependency(Fan and Lu 2019) rules to transfer expert rules to graphs. Another work(X. Li et al. 2019) extends the GNNExplainer by considering edge weights on Bitcoin OTC Data. xFraud(S. X. Rao et al. 2021) extends the GNNExplainer to heterogeneous graphs of different node and edge types. Current research cannot provide explainability on graphs from multiple views.

The above analysis shows that the existing research cannot meet the challenges we laid out Section 1. In addition, no approach is available to address the three key required functions of fraud detection in supply chain finance (see Table [tab:freq]). They should be able to model different types of information via heterogeneous graphs, share learning via multitasking, and provide explainability to decisions.

Methodology

This section presents the technical details of our proposed model MultiFraud. The MultiFraud framework is shown in Fig. [fig:archi]. 1) Multi-view representation learning: utilizing heterogeneous graph neural networks to learn entity embeddings. 2) Multi-entity embedding sharing: utilizing an attention-based component to share embedding between entities. 3) Multi-task fraud detector: utilizing multitask learning to train fraud detection of multiple entities jointly. 4) Multi-view fraud explanation: providing node and edge explanation across multiple views.



Heterogeneous Graph Construction

We construct two heterogeneous graphs consisting of critical entities in supply chain finance: enterprises and transactions. Given enterprise graph $\mathcal{G}_e = \left\{ \{\mathcal{V}_p | p \in \mathcal{P}\}, \{\mathcal{E}_q | q \in \mathcal{Q}\} \right\}$ with $|\mathcal{P}|$ types of nodes and $|\mathcal{Q}|$ types of edges. $\mathcal{V}_p$ is the node set of type p and $\mathcal{E}_q$ is the edge set of type q. The node type set $\mathcal{P} = \{e, l, s, x, t, m\}$ in the heterogeneous enterprise graph includes nodes from enterprise, legal person, shareholder, executive, telephone and email, respectively; and the edge type set is $\mathcal{Q} = \{ee, el, es, ex, et, em\}$, where $\mathcal{E}_{ee}$ includes all edges built through the enterprise being a shareholder of another enterprise. Other edge types represent that an enterprise is related to another node type in $\{l, s, x, t, m\}$. Each enterprise node carries features containing commercial information such as registration time, registered capital, paid-up capital, judicial information such as administrative penalties and legal proceedings, and business information such as bidding, certificates, etc.

Similarly, a transaction graph $\mathcal{G}_t = \left\{ \left\{ \textcolor{blue}{\mathcal{U}_p} |p \in \mathcal{P}^{\prime} \right \},\left

\{\mathcal{E}_q^{\prime} | q \in \mathcal{Q}^{\prime} \right \} \right \}$ is constructed. In order to predict the legitimacy of each transaction, we follow the same approach as described in (C. Liu et al. 2021; S. X. Rao et al. 2021), treating each transaction as a node in the graph. The node type set $\mathcal{P}' = \{t, s, r, n, d\}$ includes transaction, sender, receiver, network information, and device information, respectively; and the edge type set is $\mathcal{Q}' = \{ts, tr, tn, td\}$. If a transaction has a relationship with another type of node in $\mathcal{P}' = \{s, r, n, d\}$, we put an edge between those two nodes. Each transaction node carries node features containing information, including transaction amount, redemption period, etc. Due to the significantly larger number of transactions compared to the number of enterprises, the scale of the constructed transaction graph can be quite large. While current graph neural network models are capable of training on large-scale graphs, such as IHGAT(C. Liu et al. 2021) and xFraud(S. X. Rao et al. 2021), which respectively utilize transaction graphs with 1.76 million and 1.1 billion nodes, for resource control in practical applications, it is possible to specify a specific time frame for the transaction graph to reduce its size. For example, one can consider only the transaction records of enterprises within the past six months.

Multi-view Representation Learning

Multi-view representation learning aims to learn node embeddings for entities using heterogeneous graph neural networks.

Multifraud is a flexible framework that allows for using different heterogeneous GNNs based on the characteristics of various entities. To validate the effectiveness of the framework, we employ metapath-based GCN models for learning. The formal definition of metapath is as follows:

Definition 1. Metapath. A metapath P is defined as a path in the form of $\mathcal{P}_1 \xrightarrow{\mathcal{Q}_1} \mathcal{P}_2 \xrightarrow{\mathcal{Q}_2} \cdots \xrightarrow{\mathcal{Q}_l} \mathcal{P}_{l+1}$(abbreviated as $\mathcal{P}_1\mathcal{P}_2 \cdots \mathcal{P}_{l+1}$), which describes a relation $\mathcal{Q} = \mathcal{Q}_1 \circ \mathcal{Q}_2 \circ \cdots \circ \mathcal{Q}_l$ between node types $\mathcal{P}_1$ and $\mathcal{P}_{l+1}$, where $\circ$ is the composition operator on relations.

For the enterprise graph $\mathcal{G}_e = \{\mathcal{V}, \mathcal{E}\}$, each enterprise node $v \in \mathcal{V}$ is associated with a feature vector x_v. We apply preprocessing using a feedforward network to the node features to generate initial node representations, i.e., $h_v^0 = FFN(x_v)$. The embedding of k-th layer is computed as follows:

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

where $\mathbf{W}_k$ is a learnable matrix, and $N(v)$ refers to the neighbors of v, σ represents the sigmoid function.

For each metapath, We apply GCN of two layers with skip connections. The output embedding $\mathbf{z}_v = \mathbf{h}_v^K$ from M different metapaths are concatenated:

$$\mathbf{z}_v^F = \bigoplus_{i=1}^{M} \mathbf{z}_v^i$$

where $\bigoplus$ represents the concatenation operation.

For each transaction node u in $\mathcal{G}_t$, we obtain $\mathbf{z}_u^F$ in the same way.

Multi-entity Embedding Sharing

To construct direct interaction between multiple views, we propose an attention-based component to share embeddings between multiple entities. Attention scores are further used in multi-view explanations. Different attention-based methods can be applied based on the characteristics of different entities.

For transactions with time series, LSTM is a popular method to capture sequential information. To generate attention scores for every transaction, we apply attention-based bidirectional LSTM to learn the representation. We define the transaction history of the enterprise as $T = \{txn_1, txn_2, \dots, txn_t\}$. The initial embeddings are $\mathbf{Z}_{uT}^F = \{z_{u1}^F, z_{u2}^F, \dots, z_{ut}^F\}$. The LSTM layer computes as follows:

$$i_t = \sigma(W_{ii} z_{ut}^F + b_{ii} + W_{hi} h_{t-1} + b_{hi})$$
$$f_t = \sigma\left(W_{if} z_{ut}^F + b_{if} + W_{hf} h_{t-1} + b_{hf}\right)$$
$$g_t = \tanh\left(W_{ig} z_{ut}^F + b_{ig} + W_{hg} h_{t-1} + b_{hg}\right)$$
$$o_t = \sigma(W_{io} z_{ut}^F + b_{io} + W_{ho} h_{t-1} + b_{ho})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$
$$h_t = o_t \odot \tanh(c_t)$$

where z_{ut}^F is the input, h_t is the hidden state at time t, h_{t-1} is the hidden state of the layer at time $t-1$ or the initial hidden state, c_t is the cell state, and i_t, f_t, g_t, o_t are the input, forget, cell, and output gates, respectively. W are learnable weight matrices, and $\odot$ represents the Hadamard product.

The transaction sequences are processed in both directions:

$$h_i = \left[\overrightarrow{h_\imath} + \overleftarrow{h_\imath}\right]$$

The attention score α_i is calculated through Softmax. We record the attention scores as $\mathbf{Att}_v$ for explanation. The output is the weighted sum of the hidden states of all transactions:

$$u_i = \tanh(W_a h_i + b_a),$$
$$\alpha_i = \frac{\exp(u_i)}{\sum_i \exp\,(u_i)},$$
$$\mathbf{z}_v^A = \sum_i \alpha_i\, h_i$$

The historical transaction embedding $\mathbf{z}_v^A$ is concatenated with $\mathbf{z}_v^F$:

$$\mathbf{z}_v^{new} = \mathbf{z}_v^F \oplus \mathbf{z}_v^A$$

For enterprises, we treat the enterprises involved in transactions equally, i.e., the attention scores are the same for enterprises. The embeddings of involved enterprises are concatenated with the embedding of the transaction $\mathbf{z}_u^F$:

$$\mathbf{z}_u^{new} = \mathbf{z}_u^F \oplus \mathbf{z}_{vs}^F \oplus \mathbf{z}_{vr}^F$$

We apply post-processing using a feedforward network to $\mathbf{z}_v^{new}$ and $\mathbf{z}_u^{new}$ to generate the final node embeddings separately and feed them into a Softmax layer to predict the node class.

We use Algorithm 1 to illustrate the detailed process of the detector. The codes in line 1 and 2 process the input features using feedforward networks. The codes between 3 and 14 learn the embeddings of enterprise and transaction nodes. GCN is adopted on every metapath to generate embeddings. The embeddings are then concatenated. The codes between 15 and 17 concatenate embeddings of enterprise nodes with transaction embeddings generated by attention-based bidirectional LSTM. The codes between 18 and 20 concatenate embeddings of transaction nodes with enterprise embeddings. The outputs are fed to the feedforward network and Softmax activation function to normalize them into a probability distribution to support decision-making.

$h_v^0 = FFN(x_v)$

$h_u^0 = FFN(x_u)$

$\mathbf{h}_v \leftarrow \sigma\left(\mathbf{W} \sum_{a \in N(v) \cup v} \frac{\mathbf{h}_{va}}{\sqrt{|N(a)||N(v)|}}\right)\ \mathbf{z}_v^F \leftarrow \oplus_{i=1}^{M} \mathbf{z}_v^i\,;$

$\mathbf{h}_u \leftarrow \sigma\left(\mathbf{W}' \sum_{b \in N(u) \cup u} \frac{\mathbf{h}_{ub}}{\sqrt{|N(b)||N(v)|}}\right);\ \mathbf{z}_u^F \leftarrow \oplus_{i=1}^{M'} \mathbf{z}_u^i\,;$

$\mathbf{z}_v^{new} \leftarrow \mathbf{z}_v^F \oplus Attention - BLSTM(z_{u1}^F, z_{u2}^F, \dots, z_{ut}^F)$

$\mathbf{z}_u^{new} \leftarrow \mathbf{z}_u^F \oplus \mathbf{z}_{vs}^F \oplus \mathbf{z}_{vr}^F$

$S_v = Softmax(FFN(\mathbf{z}_v^{new}))$

$S_u = Softmax(FFN(\mathbf{z}_u^{new}))$

Multi-task Fraud Detector

We use the supervised standard cross-entropy loss for node predictions of entities. The loss functions for enterprises and transactions are defined as follows:

$$\begin{aligned} \mathcal{L}_{enterprise} &= -\sum_{v \in \mathcal{V}_{vt}} [y_v \log(S_v) + (1 - y_v)\log(1 - S_v)] \\ \mathcal{L}_{transaction} &= -\sum_{u \in \mathcal{V}_{ut}} [y_u \log(S_u) + (1 - y_u)\log(1 - S_u)] \end{aligned}$$

where $\mathcal{V}_{vt}$ and $\mathcal{V}_{ut}$ are the training sets of enterprise nodes and transaction nodes, y_v and y_u are the labels of enterprise node v and transaction node u.

Different tasks are related to each other. The model shares representations of different entities between tasks. Therefore, we use the multi-task learning method to train the two tasks jointly. The combined loss function is defined as follows:

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

where λ is used to balance the importance of tasks. The default setting of λ is 0.5.

Multi-view Fraud Explanation

Under the multi-view setting, the explainer is responsible for explaining on multiple entities. We construct a multi-view fraud explainer based on attention and GNNExplainer(Ying et al. 2019). GNNExplainer is a model-agnostic approach for providing explanations of GNN's predictions(Ying et al. 2019).

Given Φ as the trained fraud detection model, the prediction $\hat{y}$ is given by $\Phi(G_c(v), X_c(v), I_c(v))$, where $G_c(v)$ represents structure information, $X_c(v)$ represents feature information and $I_c(v)$ represents information from other views. The multi-view fraud explainer component generates explanations as follows:

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

where G_{vS} represents the subgraph of computation graph(Ying et al. 2019) containing the node for prediction, G_{uS} are subgraphs containing correlated nodes from other graphs. X_{vS}^F and X_{uS}^F are masked node features by F, i.e., $X_{vS}^F = \left\{x_j^F \mid v_j \in G_{vS}\right\}, X_{uS}^F = \left\{x_j^F \mid v_j \in G_{uS}\right\}$).

The explainer inputs the node index of the node-to-explain, enterprise node features, transaction node features, enterprise edge indexes, transaction edge indexes, enterprise edge types, and transaction edge types. Also, since GNNexplainer changes the original node index and our model needs to track the relationship between enterprise nodes and transaction nodes, the node mappings of enterprises and transactions are used as input. The output includes node feature masks and edge masks from two graphs.

For enterprises, the explainer is applied to the enterprise node first to generate X_{vS}^F and G_{vS} of the enterprise. Secondly, due to the long transaction sequence of enterprises, providing too much explanation content will increase the complexity of understanding. Therefore, rather than providing explanations for every transaction, we provide the top H transactions with the highest attention scores(Vaswani et al. 2017). Then we apply the explainer for these transactions on the transaction graph to generate $\cup_{i\in k} X_{uSi}^F$ and $\cup_{i\in H} G_{uSi}$. The subgraphs derived from different views are interconnected to produce the ultimate explanation result.

For transactions, the explainer is applied to the transaction node first to generate important features X_{uS}^F and the subgraph G_{uS} of the transaction. Then the explainer is applied on the enterprise graph to generate meaningful features $\cup_{i\in 2} X_{vSi}^{F}$ and subgraphs $\cup_{i\in 2} G_{vSi}$.

Algorithm 2 illustrates the detailed process of the explainer. In line 1, the explainer generates node feature masks and edge masks from the perspective of enterprise or transaction first. Related transactions(line 2 and 3) or enterprises(line 4 and 5) are picked based on the type of node-to-explain. Then the explainer generates node feature masks and edge masks from the other perspective(line 6 to 8). The edge masks are connected to form the final explanation subgraph(line 9 to 11).

$G_{vS}, X_{vS}^{F} \leftarrow GNNExplainer(v, D)$

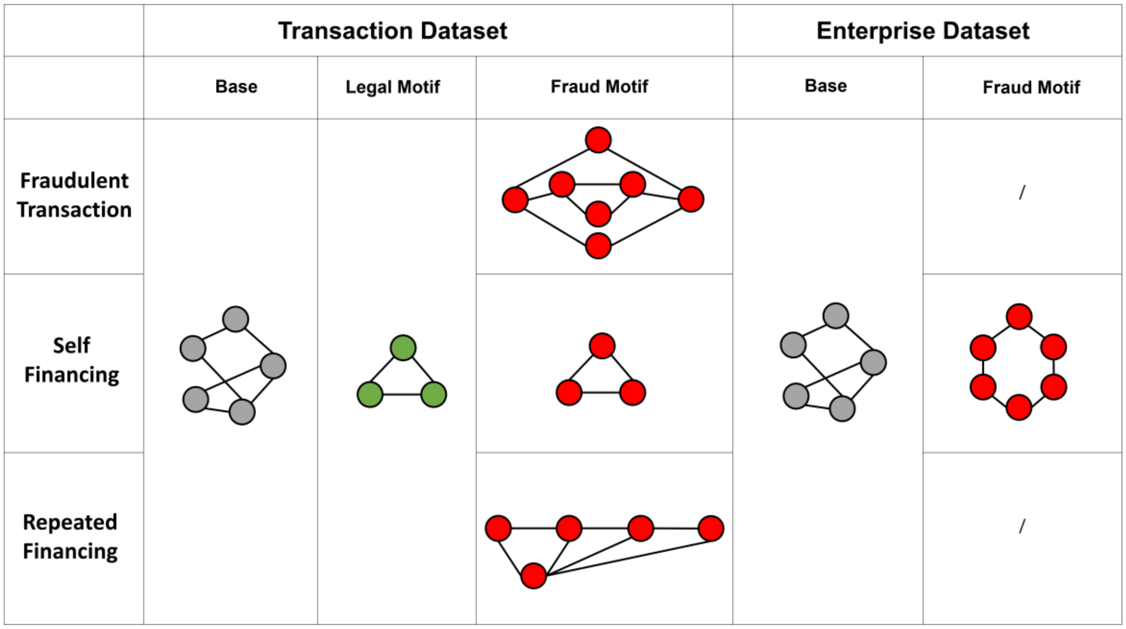
Experiment

To test the effectiveness of MultiFraud, we designed experiments on five datasets. We first introduce four synthetic datasets and a real-world dataset. Then, the comparing methods, experimental settings and implementation details are provided. We conduct experiments and case studies to answer the following questions:

- Question 1: Can MultiFraud outperform the up-to-date and state-of-the-art baseline algorithms on fraud detection?
- Question 2: Can the multitasking module effectively improve the performance of fraud detection?
- Question 3: Can MultiFraud derive meaningful explanations from multiple graphs?

Datasets

We provide four synthetic and one real-world datasets to test whether our proposed framework is effective in fraud detection and explanation. We also use these datasets as benchmarks to compare our approach with existing ones.



	Transaction Dataset			Enterprise Dataset	
	Base	Legal Motif	Fraud Motif	Base	Fraud Motif
Fraudulent Transaction					/
Self Financing					
Repeated Financing					/

Synthetic dataset

From investigating historical cases, government policies[2] have identified that the common types of supply chain finance fraud are fraudulent transactions, self-financing, and repeated financing. We design the corresponding synthetic dataset for testing for each of the three main types of accounts receivable fraud. Meanwhile, we designed a mixed fraud dataset that contains all three fraud types. Each dataset contains an enterprise dataset and a transaction dataset.

Many observed networks are in the class of scale-free networks, including enterprise and transaction networks. The Barabási-Albert(BA) model is an algorithm for generating scale-free networks. We combine the BA model with motifs to generate synthetic datasets. The motif is used to represent a specific subgraph for transactions and enterprises in accounts receivable finance in Section 2. We first introduce the definition of the motif(Zhao et al. 2018) as follows:

Definition 2. Motif. A motif M is a subgraph defined on n nodes by a $n \times n$ binary adjacency matrix B_M.

Fig. [fig:dataset] shows the critical components for generating synthetic datasets. The grey graphs represent the base graphs for generating data. Legal motifs in transaction datasets represent the legal business flow of accounts receivable financing. Fraud motifs represent the business flow of different fraud types, which are described in detail in Section 5.4. For enterprise datasets, only self-financing fraud has a specific motif.

For transaction datasets, we start with a base BA graph. Every node in the BA graph represents a motif(business flow). We pick random nodes to expand to fraud motifs, and

[2] http://xkzj.mofcom.gov.cn/article/myszh/llyzc/202107/20210703179057.shtml

the rest are expanded to legal motifs. The structure of fraud motifs is different for different fraud types. We construct these motifs based on the examples reported in Section 5.4. The structure of fraud transaction motifs in self-financing fraud is the same as legal transaction fraud motifs. The original edges are reserved to connect expanded motifs. We add 0.09N random edges to the graph. N is the number of edges in the current graph. For enterprise datasets, we start with a base BA graph. We pick random nodes to be fraudulent enterprises. For fraudulent transaction and repeated financing types, the neighboring nodes of these fraud nodes have a 40% probability of being fraudulent. For the self-financing type, the fraud nodes are expanded to corresponding fraud motifs. We add 0.01N random edges to the graph. We construct a correlation between different entities by assigning fraudulent transactions to fraudulent enterprises and legitimate transactions to random enterprises. The features of nodes belonging to the same category are sampled from normal distributions.

Real-world dataset

We construct a real-world dataset containing enterprises with transactions. HAT(Zheng et al. 2021) is a dataset for bankruptcy prediction of 13489 enterprises in China, containing shareholder and board member relationships. BankSim(Lopez-Rojas and Axelsson 2014) is a dataset for fraud transaction prediction. We match customers with fraudulent transactions with fraudulent enterprises, while others are matched with legitimate enterprises.

Table [table:dataset] shows the statistical information describing datasets, including the number of nodes, percentage of fraudulent nodes, length of feature vectors, relations and the number of edges. E-E and T-T in synthetic datasets represent the direct relationship between enterprises nodes and the direct relationship between transactions nodes, respectively. In the HAT dataset, E-S-E represents two enterprises with the same shareholder. E-E represents direct shareholding between two enterprises. E-M-E represents two enterprises with the same board member. ALL represents the sum of the three previous relationships. In the BankSim dataset, T-A-T denotes the relationship we built based on the amount of transactions.

[table:dataset]

Comparing Methods

We evaluate the performance of fraud detection of MultiFraud by comparing with these baselines:

- GraphSage(Hamilton, Ying, and Leskovec 2017): A graph neural network that learns node representations by sampling and aggregating features from local neighborhood.

- GEM(Ziqi Liu et al. 2018): A heterogeneous GNN on account-device graphs leveraging two weaknesses of attackers: device aggregation and activity aggregation.

- SemiGNN(D. Wang et al. 2019): A heterogeneous GNN that adopts hierarchical attention mechanism for financial fraud detection.

- GraphConsis(Zhiwei Liu et al. 2020): A heterogeneous GNN that uses context embeddings, consistency filtering and relation weights in fraud detection with inconsistency problem.
- RioGNN(Peng et al. 2021): A heterogeneous GNN that uses label-aware neural similarity measure and reinforcement learning framework.
- MultiFraud-S: Ablation model that remove multi-entity embedding sharing component.

Experimental settings and implementation

We use Macro-F1 and AUC as evaluate metrics for fraud detection since the datasets are highly imbalanced. The Macro-F1 is the unweighted mean of the F1-score for legitimate and fraudulent entities. F1-score is defined as:

$$F1 = 2 \cdot \frac{\text{Precision} \ \cdot \ \text{Recall}}{\text{Precision} \ + \ \text{Recal}}$$

Macro-F1 is defined as:

$$Macro - F1 = \frac{F1_{legal} + F1_{fraud}}{2}$$

where $F1_{legal}$ is F1-score of the legal class and $F1_{fraud}$ is F1-score of the fraud class.

AUC is defined as:

$$AUC = \frac{1}{2}\sum_{i=1}^{m-1}(x_{i+1} - x_i)\,(y_i + y_{i+1})$$

where y is True Posotive Rate$\left(TPR = \frac{TP}{TP+FN}\right)$, x is False Positive Rate$\left(FPR = \frac{FP}{FP+TN}\right)$.

For MultiFraud, the hyperparameter settings are shown in [table:dual_hyperparameter]. We use Adam(Kingma and Ba 2015) for model optimization. For other baseline methods, we use grid search to tune the best parameters for different datasets.

[table:dual_hyperparameter]

The codes for RioGNN[3] are from the authors' original implementations. The codes for other models are from an open-source implementation DGFraud-TF2(Dou et al. 2020). For each dataset, we apply the same partitioning ratio, which is 68% for training, 12% for validation, and 20% for the test set. For a transaction graph with time information, after sorting the transactions in chronological order, the dataset is divided according to the set ratio. The codes of our model and the experimental datasets are accessible at

[3] https://github.com/safe-graph/RioGNN

https://github.com/MultiFraudGNN/MultiFraud. All experiments are conducted on 1 A100-SXM4-80GB(80GB) GPU, 32 vCPU AMD EPYC 7763 64-Core Processor, 240GB RAM.

Results

Fraud Types

The experiments show that the system has successfully identified three supply chain account receivable fraud types: fraudulent transactions, self-financing, and repeated financing. Fraudulent transactions refer to the falsification of information such as trade and logistics. Self-financing means that the borrower and the guarantor have close relationships such as with relatives or friends or are controlled by the borrower. Repeated financing refers to repeatedly opening multiple warehouse receipts for the same batch of goods, and obtaining loans through repeated receivables to multiple financing institutions(Hua and Xuan 2018). To introduce our motivation, we present examples of these three fraud types. For each example, we analyze from two perspectives: the business process and the relationships between enterprises, including equity and shareholders, etc. This is done to showcase the different behaviors on transaction graphs and enterprise graphs under various fraud types.

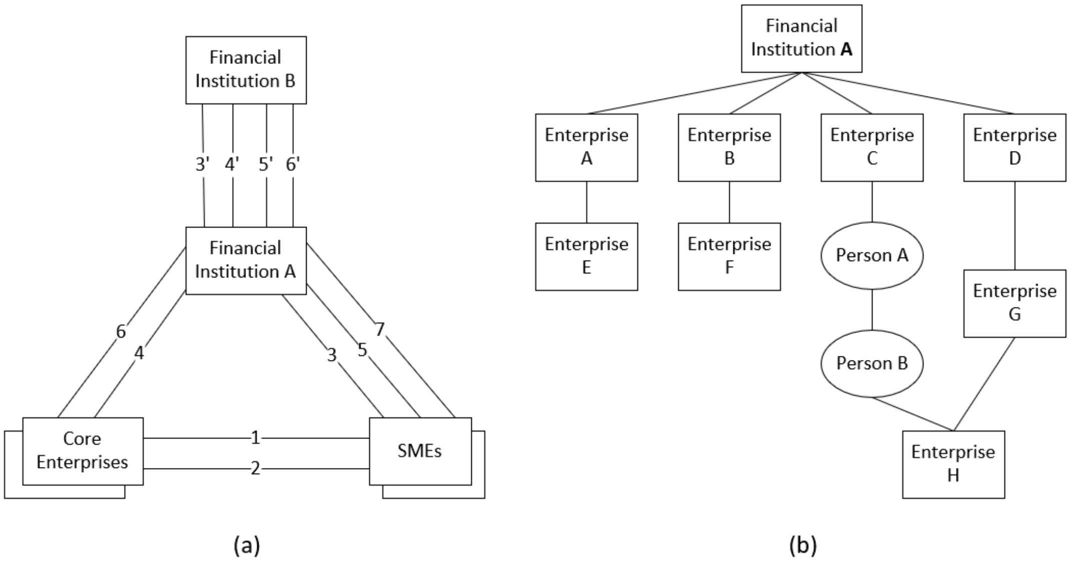
[Example 1] For fraudulent transaction fraud, we take an example of a delayed redemption event that occurred in 2021 involving over RMB 20 billion in Fig.3. The business process of the event is shown in Fig.3(a). Apart from the regular business model, Financial Institution A (a factoring company) resells the receivables to another Financial Institution B (a trust company). However, these receivables are all forged or expired. Regarding investment and equity relationships between enterprises shown in Fig.3(b), the shareholders (Enterprises A-D) of Financial Institution A are all law-abiding enterprises. However, Enterprise C's legal person (Person A) is a defaulting person. A defaulting person refers to an individual who has engaged in relevant illegal activities and has been disclosed. Person A and B have multiple co-investment relationships, and Person B is also a defaulting person. Enterprise D is related to the defaulting Enterprise H through Enterprise G.

[Example 2] For fraudulent transaction fraud, we provide another example of self-purchasing and self-selling that occurred in 2021 in Fig.4. The business process of the event is shown in Fig.4(a). The SME purchases products from another SME at a low price and then prompts the other SMEs to repurchase the products at a higher price. Through this continuous operation, they falsify accounts receivable. These SMEs are all related entities, as evidenced by the interconnected relationships in the enterprise graph in Fig.4(b).

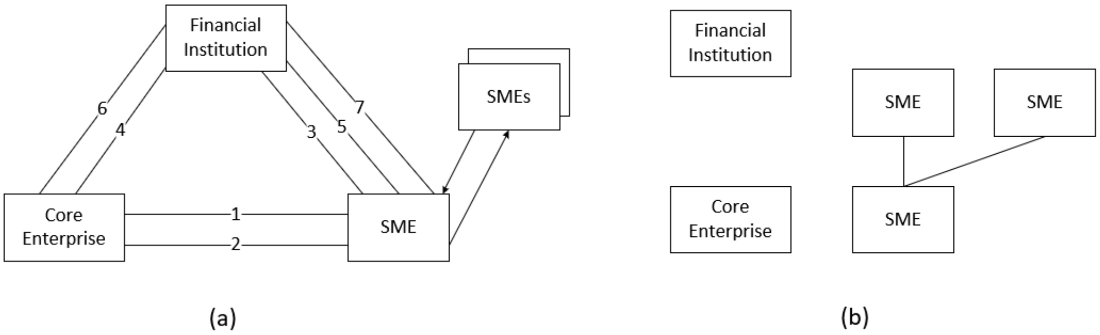
[Example 3] For self-financing fraud, we take another example that occurred in 2021 in Fig.5. The business process of the event is the same as regular events, as shown in Fig.5(a). However, the borrower (SME A), financier (Financial Institution C) and guarantor (Core Enterprise K) are related parties in the same group, as shown in Fig.5(b). SME A is a wholly owned subsidiary of Core Enterprise A. Financial Institution C is a wholly owned subsidiary of Enterprise I. Enterprise J and Enterprise I have the same legal person and chairman. Enterprise K has the same registered telephone number (Telephone A) and legal person

(Person C) as Enterprise J. Enterprise K is one of the shareholders of Core Enterprise A. The SME has the same executive (Person D) as Financial Institution C.

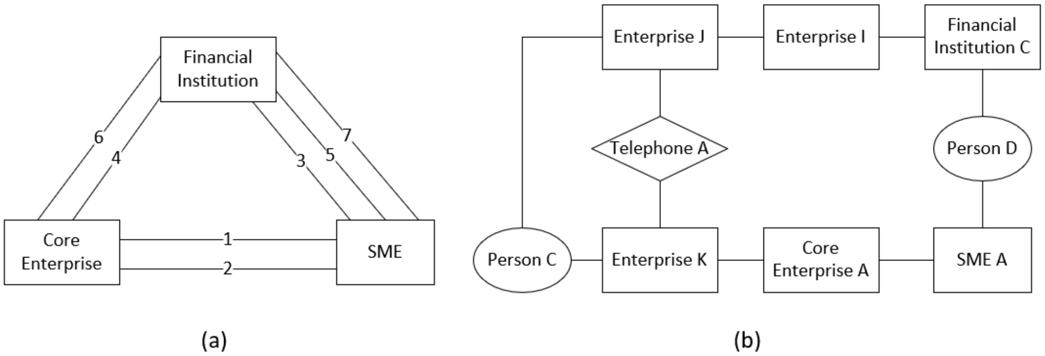
[Example 4] For repeated financing fraud, we take an example that occurred in 2014 in Fig.6. Fig.6.(a) shows that SME B uses the same receivables to apply for loans from several different financial institutions. There is no noticeable relationship between these companies in Fig.6(b).



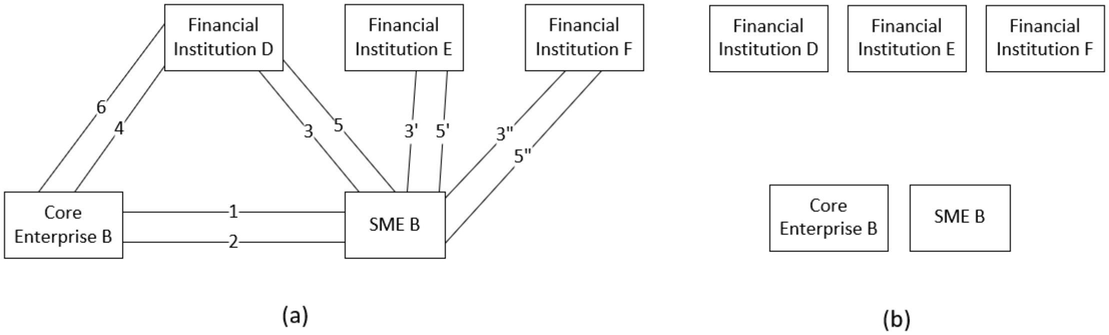
Example 1 on fraudulent transaction.



Example 2 on fraudulent transaction.



Example 3 on self-financing.



Example 4 on repeated financing.

Identifying these frauds is difficult, as fraudsters try to conceal and disguise them, and the financial information is large, complex and imbalanced. The proposed model effectively identifies different types of fraud by learning from various types of information and relationships and analyzing multiple views within relation networks.

Fraud Detection Benchmarking

To answer Questions 1 and 2, we present experimental results in Tables [tab:exp1] and [tab:exp2].

In synthetic datasets, MultiFraud outperforms other methods, including the mixed fraud dataset and the single-type fraudulent transaction, self-financing, and repeated financing datasets. The results show that MultiFraud is effective in fraud detection tasks in supply chain finance.

For baseline methods, the performance varies for different fraud types. Regarding AUC, the best-performing baseline methods for fraudulent transaction, self-financing, and repeated financing are GraphSage, GEM, and RioGNN, respectively. It indicates that different models have different capabilities in detecting different types of fraud. MultiFraud outperforms other baseline methods in all fraud types, indicating that the proposed model has better applicability in fraud detection types.

From different views, in fraud transaction detection, the GEM model achieves the best baseline effect in most fraud types, indicating that different models have different capabilities in detecting fraud of different entities. MultiFraud achieves the best effect in both enterprise and transaction fraud, indicating that the proposed model has better applicability when facing the differences in attributes and structural features of different entities.

In the real-world dataset, MultiFraud achieves improvement compared with other baseline methods. The AUC for both GraphSage and GEM is 0.5. This is because of class imbalance, which prevents them from learning meaningful features and leads them to predict all transactions as legitimate. The proposed model shares information among different entities through a combination of temporal modeling and feature concatenation. It utilizes a multitasking framework to leverage relationships between different tasks. During this process, the model utilizes attributes and structural information from different entities as auxiliary information, thereby increasing the distinctiveness of features from different categories. Incorporating auxiliary information with multitask learning(Spangher et al. 2021) proves effective in severely imbalanced real-world transaction datasets. Therefore, the proposed model mitigates the class imbalance problem to some extent.

The improvements of the proposed model on enterprise fraud detection are smaller than those on synthetic datasets. This is due to the more complex features and relationships in real-world datasets. In the experiments conducted in the literature(Zheng et al. 2021), the differences in performance among different methods were similarly not very pronounced, which aligns with the results observed in our experiments.

[tab:exp1]

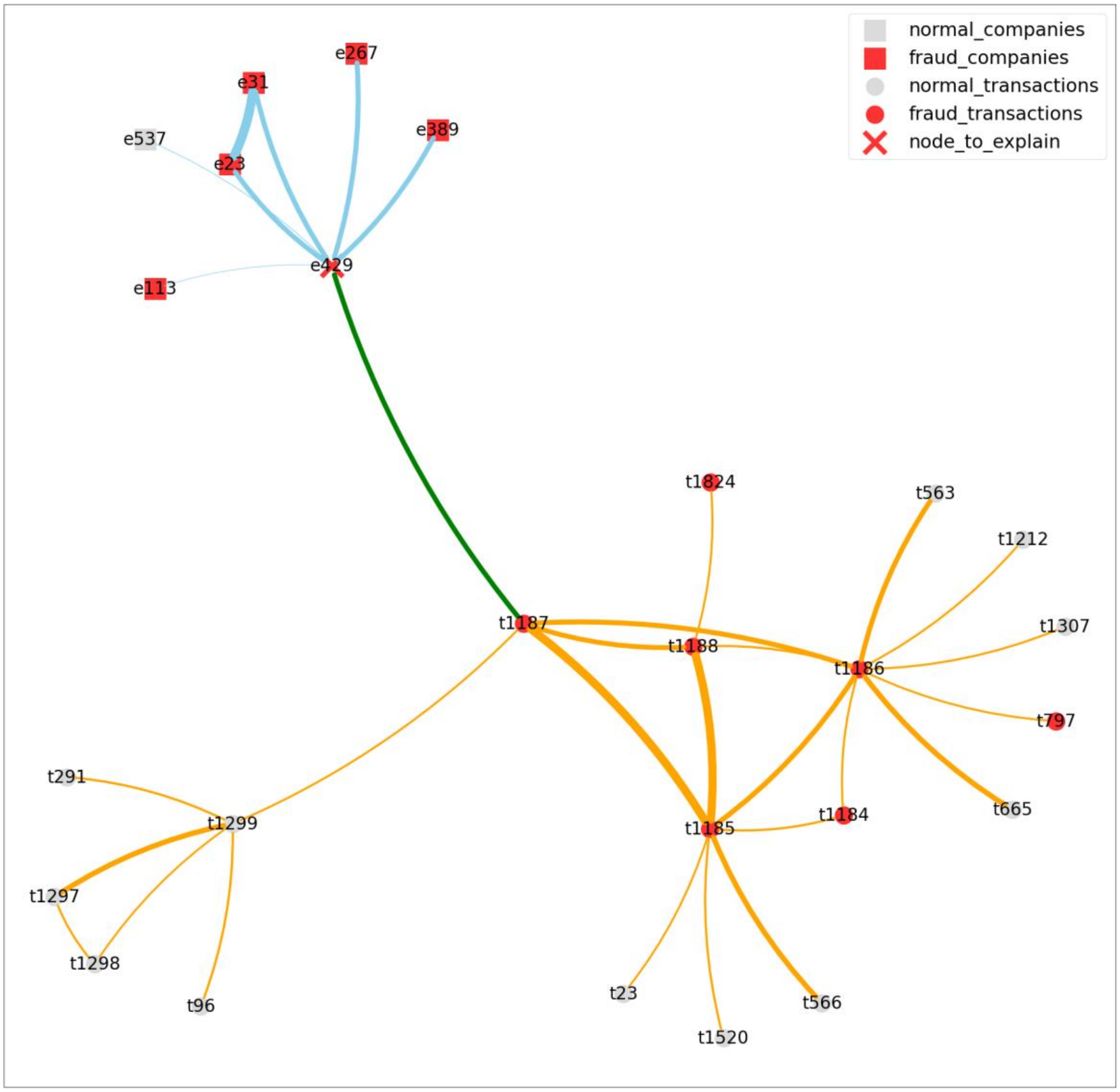
[tab:exp2]

Explainability

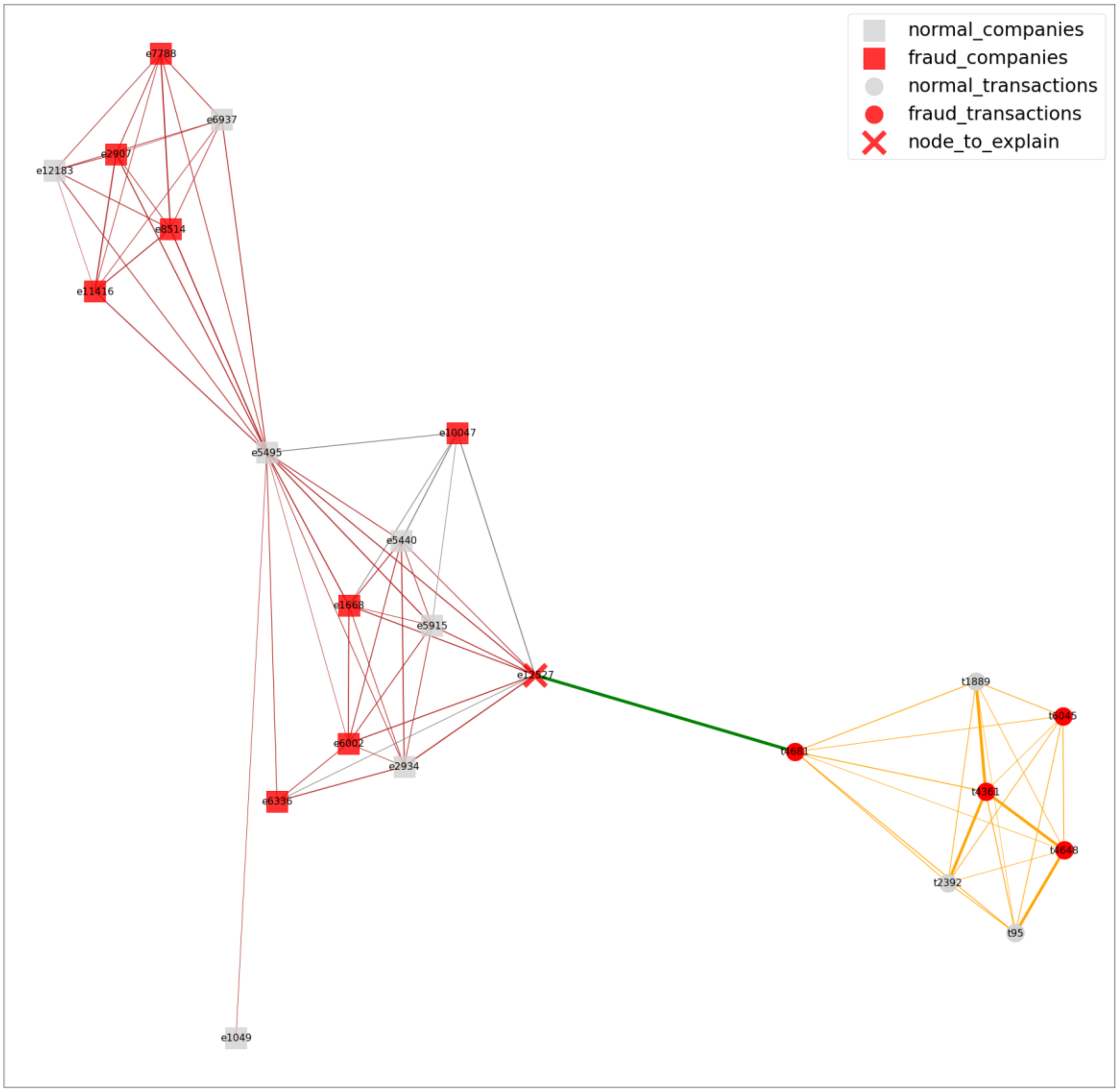
To answer Question 3, we present explanations of fraud predictions in Fig. 7 and Fig. 8. In the visualization, distinct shapes are used to denote different types of nodes, while varying colors on the edges represent different types. The thickness of the edges represents the weights, with thicker edges denoting greater importance. Additionally, the ground truth labels are represented by the color of nodes.

We visualize an explanation for node *e429* within the synthetic dataset in Figure 7, which is an enterprise that engages in repeated financing fraud. The explainer effectively captures the majority of crucial edges connecting it to other fraudulent enterprises in its neighborhood. Additionally, the generated transaction (node *t1187*) with the highest score in **Att** is fraudulent. The explainer component also identifies the most suspicious edges related to transactions, successfully pinpointing critical edges within the motif.

We visualize another explanation for a enterprise within the real-world dataset in Figure 8. For shareholder relationships, the enterprises connected are all fraudulent. For board member relationships, only one connected enterprise*e1668* is fraudulent. Despite enterprise *e5495* being legal, it connects with several fraudulent enterprises through board member relationship. From the transaction perspective, the transaction chosed is fraudulent, while connected with both legitimate and fraudulent transactions. These examples showcase MultiFraud's capability to identify suspicious entities, even in scenarios where the fraud is concealed by legitimate entities.



Visualization of fraud explanation in the synthetic dataset.



Visualization of fraud explanation in the real-world dataset.

Conclusion

We propose an end-to-end multitask framework, MultiFraud, to provide effective fraud prediction in supply chain finance, despite their complexity and concealment to evade detection. The strength of MultiFraud is that it can explore the supply chain financing business model to construct different heterogeneous graphs of enterprises and transactions and capture their characteristics. The multitask framework possesses the flexibility to learn the embeddings of both enterprise and transaction domains and effectively exploit their correlation. It utilizes atteion-based model to process transaction sequences and generate importance weights, which are used in the explainer component to extend graph explainability to multiple graphs to provide a complete picture. The experimental results demonstrate the effectiveness of correctly detecting fraud and provide sufficient explanations for the derived conclusions. We also compare with other existing GNN methods based on the test of five datasets. MultiFraud has outperformed them on fraud detection by a large margin in all fraud types and domains, according to the criteria Macro-F1 and AUC.

Supply chain finance contains a large amount of information to be explored, including multimodal data such as text(Xie et al. 2022), images, and videos. As a flexible framework, graph representation learning with more complex models can be studied in the future.

Acknowledgment

Thanks to our industry partner Ouyeel. Thanks to Xinyu Yao, Zhenzhen Gu and Boyan Zhang for their help on the case study.

References

Aboutorab, Hamed, Omar K. Hussain, Morteza Saberi, Farookh Khadeer Hussain, and Elizabeth Chang. 2021. "A Survey on the Suitability of Risk Identification Techniques in the Current Networked Environment." *JNCA* 178: 102984. https://doi.org/10.1016/j.jnca.2021.102984.

Albashrawi, Mousa. 2021. "Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015." *J Data Sci.* 14 (3): 553–70. https://doi.org/10.6339/JDS.201607_14(3).0010.

Barredo Arrieta, Alejandro, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, et al. 2020. "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges Toward Responsible AI." *Inf Fusion* 58: 82–115.

Bussmann, Niklas, Paolo Giudici, Dimitri Marinelli, and Jochen Papenbrock. 2020. "Explainable AI in Fintech Risk Management." *Front. Artif. Intell.* 3: 26.

Chicha, Elie, Bechara Al Bouna, Kay Wünsche, and Richard Chbeir. 2021. "Exposing Safe Correlations in Transactional Datasets." *Service Oriented Computing and Applications* 15 (4): 289–307. https://doi.org/10.1007/s11761-021-00325-1.

Conforti, Raffaele, Marcello La Rosa, Giancarlo Fortino, Arthur HM Ter Hofstede, Jan Recker, and Michael Adams. 2013. "Real-Time Risk Monitoring in Business Processes: A Sensor-Based Approach." *Journal of Systems and Software* 86 (11): 2939–65.

Constante-Nicolalde, Fabián-Vinicio, Paulo Guerra-Terán, and Jorge-Luis Pérez-Medina. 2020. "Fraud Prediction in Smart Supply Chains Using Machine Learning Techniques." In *Applied Technologies*, 145–59.

Deng, Aimin, and Bo Yu. 2017. "Research Overview of Risk Management of SMEs Accounts Receivable Financing Based on Supply Chain Finance." *World Journal of Research and Review* 4 (1): 16–20.

Dou, Yingtong, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. 2020. "Enhancing Graph Neural Network-Based Fraud Detectors Against Camouflaged Fraudsters." In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, edited by Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux, 315–24. ACM. https://doi.org/10.1145/3340531.3411903.

Fan, Wenfei, and Ping Lu. 2019. "Dependencies for Graphs." *ACM Trans. Database Syst.*

Hamilton, William L., Zhitao Ying, and Jure Leskovec. 2017. "Inductive Representation Learning on Large Graphs." In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, edited by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, 1024–34.

Hassan, Mohammad Mehedi, Abdu Gumaei, Ahmed Alsanad, Majed Alrubaian, and Giancarlo Fortino. 2020. "A Hybrid Deep Learning Model for Efficient Intrusion Detection in Big Data Environment." *Information Sciences* 513: 386–96.

He, Xiangjun, and Lingyun Tang. 2012. "Exploration on Building of Visualization Platform to Innovate Business Operation Pattern of Supply Chain Finance." *Physics Procedia*, ICMPBE, 33: 1886–93.

Hu, Haiju, Yakun Li, Mao Tian, and Xinjiang Cai. 2022. "Evolutionary Game of Small and Medium-Sized Enterprises' Accounts-Receivable Pledge Financing in the Supply Chain." *Syst.* 10 (1): 21.

Hua, Song, and Yang Xuan. 2018. "Risk Source and Systematic Management of Supply Chain Finance: An Integrative Framework." *Journal of Renmin University of China* 32 (4): 119.

Jiang, Nan, Fuxian Duan, Honglong Chen, Wei Huang, and Ximeng Liu. 2021. "MAFI: GNN-based Multiple Aggregators and Feature Interactions Network for Fraud Detection over Heterogeneous Graph." *IEEE Trans. Big Data*, 1–1. https://doi.org/10.1109/TBDATA.2021.3132672.

Katz, Norman A. 2016. *Detecting and Reducing Supply Chain Fraud*. Routledge.

Kingma, Diederik P., and Jimmy Ba. 2015. "Adam: A Method for Stochastic Optimization." In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, edited by Yoshua Bengio and Yann LeCun. http://arxiv.org/abs/1412.6980.

Koh, SC Lenny, Mehmet Demirbag, Erkan Bayraktar, Ekrem Tatoglu, and Selim Zaim. 2007. "The Impact of Supply Chain Management Practices on Performance of SMEs." *Ind. Manag. Data Syst.*

Li, Ao, Zhou Qin, Runshi Liu, Yiqun Yang, and Dong Li. 2019. "Spam Review Detection with Graph Convolutional Networks." In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, edited by Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu, 2703–11. ACM. https://doi.org/10.1145/3357384.3357820.

Li, Peibo, Yixing Yang, Maurice Pagnucco, and Yang Song. 2022. “Explainability in Graph Neural Networks: An Experimental Survey.” *ArXiv Preprint* abs/2203.09258. https://arxiv.org/abs/2203.09258.

Li, Xiaoxiao, João Saúde, P. Reddy, and M. Veloso. 2019. “Classifying and Understanding Financial Data Using Graph Neural Network.” In *AAAI Workshop*.

Li, Zhenchuan, Mian Huang, Guanjun Liu, and Changjun Jiang. 2021. “A Hybrid Method with Dynamic Weighted Entropy for Handling the Problem of Class Imbalance with Overlap in Credit Card Fraud Detection.” *Expert Syst. Appl.* 175: 114750.

Liang, Ting, Guanxiong Zeng, Qiwei Zhong, Jianfeng Chi, Jinghua Feng, Xiang Ao, and Jiayu Tang. 2021. “Credit Risk and Limits Forecasting in E-Commerce Consumer Lending Service via Multi-view-aware Mixture-of-experts Nets.” In *WSDM*, 229–37.

Liu, Can, Li Sun, Xiang Ao, Jinghua Feng, Qing He, and Hao Yang. 2021. “Intention-Aware Heterogeneous Graph Attention Networks for Fraud Transactions Detection.” In *KDD*, 3280–88. https://doi.org/10.1145/3447548.3467142.

Liu, Yang, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2021. “Pick and Choose: A GNN-based Imbalanced Learning Approach for Fraud Detection.” In *WWW*, 3168–77. https://doi.org/10.1145/3442381.3449989.

Liu, Zhiwei, Yingtong Dou, Philip S. Yu, Yutong Deng, and Hao Peng. 2020. “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, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, edited by Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu, 1569–72. ACM. https://doi.org/10.1145/3397271.3401253.

Liu, Ziqi, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, Le Song, and Yuan Qi. 2019. “GeniePath: Graph Neural Networks with Adaptive Receptive Paths.” In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, the Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, the Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 4424–31. AAAI Press. https://doi.org/10.1609/aaai.v33i01.33014424.

Liu, Ziqi, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. 2018. “Heterogeneous Graph Neural Networks for Malicious Account Detection.” In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, edited by Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, et al., 2077–85. ACM. https://doi.org/10.1145/3269206.3272010.

Lopez-Rojas, Edgar Alonso, and Stefan Axelsson. 2014. “Banksim: A Bank Payments Simulator for Fraud Detection Research.” In *EMSS*, 144–52.

Lu, Yi-Ju, and Cheng-Te Li. 2020. “GCAN: Graph-Aware Co-Attention Networks for Explainable Fake News Detection on Social Media.” In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 505–14. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.48.

Luo, Dongsheng, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. “Parameterized Explainer for Graph Neural Network.” In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual*, edited by Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin.

Malhotra, Arvind, Sanjay Gosain, and Omar A. El Sawy. 2005. “Absorptive Capacity Configurations in Supply Chains: Gearing for Partner-Enabled Market Knowledge Creation.” *MIS Quarterly*, 145–87.

Peng, Hao, Hongfei Wang, Bowen Du, Md Zakirul Alam Bhuiyan, Hongyuan Ma, Jianwei Liu, Lihong Wang, et al. 2020. “Spatial Temporal Incidence Dynamic Graph Neural Networks for Traffic Flow Forecasting.” *Information Sciences* 521: 277–90. https://doi.org/10.1016/j.ins.2020.01.043.

Peng, Hao, Ruitong Zhang, Yingtong Dou, Renyu Yang, Jingyi Zhang, and Philip S. Yu. 2021. “Reinforced Neighborhood Selection Guided Multi-Relational Graph Neural Networks.” *TOIS* 40 (4): 1–46.

Ran, Hongyan, Caiyan Jia, Pengfei Zhang, and Xuanya Li. 2022. “MGAT-ESM: Multi-channel Graph Attention Neural Network with Event-Sharing Module for Rumor Detection.” *Information Sciences* 592: 402–16. https://doi.org/10.1016/j.ins.2022.01.036.

Rao, Susie Xi, Shuai Zhang, Zhichao Han, Zitao Zhang, Wei Min, Zhiyao Chen, Yinan Shan, Yang Zhao, and Ce Zhang. 2021. “xFraud: Explainable Fraud Transaction Detection.” *Proc. VLDB Endow.* 15 (3): 427–36. https://doi.org/10.14778/3494124.3494128.

Rao, Yizhuo, Xianya Mi, Chengyuan Duan, Xiaoguang Ren, Jiajun Cheng, Yu Chen, Hongliang You, Qiang Gao, Zhixian Zeng, and Xiao Wei. 2021. “Know-GNN: An Explainable Knowledge-Guided Graph Neural Network for Fraud Detection.” In *ICONIP*, 159–67.

Somasundaram, Akila, and Srinivasulu Reddy. 2019. “Parallel and Incremental Credit Card Fraud Detection Model to Handle Concept Drift and Data Imbalance.” *Neural. Comput. Appl.* 31 (1): 3–14.

Song, Hua. 2021. “Risk Management in Intelligent Supply Chain Finance.” In *Smart Supply Chain Finance*, edited by Hua Song, 587–616. Singapore: Springer Nature.

Spangher, Alexander, Jonathan May, Sz-rung Shiang, and Lingjia Deng. 2021. “Multitask Learning for Class-Imbalanced Discourse Classification.” *ArXiv Preprint*. https://arxiv.org/abs/2101.00389.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, edited by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, 5998–6008.

Wan, Fang. 2021. "XGBoost Based Supply Chain Fraud Detection Model." In *ICBAIE*, 355–58.

Wang, Chen, Yingtong Dou, Min Chen, Jia Chen, Zhiwei Liu, and Philip S. Yu. 2021. "Deep Fraud Detection on Non-attributed Graph." In *Big Data*, 5470–73.

Wang, Daixin, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. 2019. "A Semi-Supervised Graph Attentive Network for Financial Fraud Detection." In *ICDM*, 598–607. https://doi.org/10.1109/ICDM.2019.00070.

Wang, Jianian, Sheng Zhang, Yanghua Xiao, and Rui Song. 2021. "A Review on Graph Neural Network Methods in Financial Applications." *ArXiv Preprint* abs/2111.15367. https://arxiv.org/abs/2111.15367.

Wang, Jianyu, Rui Wen, Chunming Wu, Yu Huang, and Jian Xion. 2019. "FdGars: Fraudster Detection via Graph Convolutional Networks in Online App Review System." In *WWW*, 310–16. https://doi.org/10.1145/3308560.3316586.

West, Jarrod, and Maumita Bhattacharya. 2016. "Intelligent Financial Fraud Detection: A Comprehensive Review." *Comput. Secur.* 57: 47–66. https://doi.org/10.1016/j.cose.2015.09.005.

Wu, Chenyang, Jinyue Liu, and Hongmei Zhang. 2021. "Data Ecology and Accurate Portrait: Optimization of Credit Risk System for SMEs in Supply Chain Finance Based on Big Data Technology." *J. Risk Anal. Crisis Response* 11 (4).

Xie, Tingyu, Shuting Tao, Qi Li, Hongwei Wang, and Yihong Jin. 2022. "A Lattice LSTM-based Framework for Knowledge Graph Construction from Power Plants Maintenance Reports." *Service Oriented Computing and Applications*. https://doi.org/10.1007/s11761-022-00338-4.

Xiong, Xin, XunKai Li, YouPeng Hu, YiXuan Wu, and Jian Yin. 2022. "Handling Information Loss of Graph Convolutional Networks in Collaborative Filtering." *Information Systems* 109: 102051. https://doi.org/10.1016/j.is.2022.102051.

Yan, Bo, Zhuo Chen, Chang Yan, Zhenyu Zhang, and Hanwen Kang. 2021. "Evolutionary Multiplayer Game Analysis of Accounts Receivable Financing Based on Supply Chain Financing." *Int. J. Prod. Res.* 0 (0): 1–19. https://doi.org/10.1080/00207543.2021.1976432.

Yang, Shuang, and Xuesong Cai. 2022. "Bilateral Knowledge Graph Enhanced Online Course Recommendation." *Information Systems* 107: 102000. https://doi.org/10.1016/j.is.2022.102000.

Yang, Shuo, Zhiqiang Zhang, Jun Zhou, Yang Wang, Wang Sun, Xingyu Zhong, Yanming Fang, Quan Yu, and Yuan Qi. 2020. "Financial Risk Analysis for SMEs with Graph-Based Supply Chain Mining." In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, edited by Christian Bessiere, 4661–67. ijcai.org. https://doi.org/10.24963/ijcai.2020/643.

Ying, Zhitao, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. "GNNExplainer: Generating Explanations for Graph Neural Networks." In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, edited by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, 9240–51.

Yuan, Hao, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. "XGNN: Towards Model-Level Explanations of Graph Neural Networks." In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, edited by Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, 430–38. ACM. https://dl.acm.org/doi/10.1145/3394486.3403085.

Yuan, Hao, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2020. "Explainability in Graph Neural Networks: A Taxonomic Survey." *ArXiv Preprint* abs/2012.15445. https://arxiv.org/abs/2012.15445.

Yuan, Hao, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. "On Explainability of Graph Neural Networks via Subgraph Explorations." In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, edited by Marina Meila and Tong Zhang, 139:12241–52. Proceedings of Machine Learning Research. PMLR. http://proceedings.mlr.press/v139/yuan21c.html.

Zhang, Shijie, Hongzhi Yin, Tong Chen, Quoc Viet Hung Nguyen, Zi Huang, and Lizhen Cui. 2020. "GCN-Based User Representation Learning for Unifying Robust Recommendation and Fraudster Detection." In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, edited by Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu, 689–98. ACM. https://doi.org/10.1145/3397271.3401165.

Zhao, Huan, Xiaogang Xu, Yangqiu Song, Dik Lun Lee, Zhao Chen, and Han Gao. 2018. "Ranking Users in Social Networks with Higher-Order Structures." In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, edited by Sheila A. McIlraith and Kilian Q. Weinberger, 232–40. AAAI Press. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16122.

Zheng, Yizhen, Vincent C. S. Lee, Zonghan Wu, and Shirui Pan. 2021. "Heterogeneous Graph Attention Network for Small and Medium-Sized Enterprises Bankruptcy Prediction." In *KDD*, 140–51. https://doi.org/10.1007/978-3-030-75762-5_12.

Zhou, Hangjun, Guang Sun, Sha Fu, Xiaoping Fan, Wangdong Jiang, Shuting Hu, and Lingjiao Li. 2020. “A Distributed Approach of Big Data Mining for Financial Fraud Detection in a Supply Chain.” *CMC* 64 (2): 1091–1105.

Zhu, Yadong, Xiliang Wang, Qing Li, Tianjun Yao, and Shangsong Liang. 2021. “BotSpot++: A Hierarchical Deep Ensemble Model for Bots Install Fraud Detection in Mobile Advertising.” *ACM Trans. Inf. Syst.* 40 (3): 50:1–28. https://doi.org/10.1145/3476107.