Information Sciences 659 (2024) 120081

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Combining intra-risk and contagion risk for enterprise bankruptcy prediction using graph neural networks

Shaopeng Wei [a], Jia Lv [b], Yu Guo [c], Qing Yang [c], Xingyan Chen [c], Yu Zhao [c,*], Qing Li [c], Fuzhen Zhuang [d,e], Gang Kou [a,*]

[a] School of Business Administration, Faculty of Business Administration, Southwestern University of Finance and Economics, Chengdu, 611130, China
[b] Graduate School, Southwestern University of Finance and Economics, Chengdu, China
[c] Fintech Innovation Center, Financial Intelligence and Financial Engineering Key Laboratory, Southwestern University of Finance and Economics, Chengdu, China
[d] Institute of Artificial Intelligence, Beihang University, Beijing 100191, China
[e] SKLSDE, School of Computer Science, Beihang University, Beijing 100191, China

ARTICLE INFO

Keywords:
Enterprise bankruptcy prediction
Intra-risk
Contagion risk
Hyper-GNNs
Heterogeneous GNNs

ABSTRACT

Predicting the bankruptcy risk of small and medium-sized enterprises (SMEs) is crucial for making decisions about loans. Existing studies in both finance and AI research fields, however, tend to only consider either the intra-risk or contagion risk of enterprises, ignoring their interactions and combinatorial effects. This study for the first time considers both types of risk and their joint effects in bankruptcy prediction. Specifically, we first propose an enterprise intra-risk encoder based on statistically significant enterprise risk indicators for its intra-risk learning. Then, we propose an enterprise contagion risk encoder based on an enterprise knowledge graph for its contagion risk embedding. In particular, the contagion risk encoder includes both the newly proposed Heterogeneous Hyper-Graph Neural Networks (HHGNN) and Hierarchical Graph Transformer Networks (HGTN). Using these two types of encoders, we design a unified framework to simultaneously capture intra-risk and contagion risk for bankruptcy prediction. To evaluate the model, we collect real-world multi-sources data on SMEs and build a novel benchmark dataset called SMEsD. We provide open access to the dataset, which is expected to further promote research on financial risk analysis. Experiments on SMEsD against twelve state-of-the-art baselines demonstrate the effectiveness of the proposed model for bankruptcy prediction.

1. Introduction

Small and medium-sized enterprises (SMEs) contribute up to 40% of gross domestic product (GDP) in emerging economies and provide more than 50% of employment worldwide.[1] Predicting the financial risk of SMEs is of great importance for both government policymakers and financial institutions. Previous studies of enterprise risk in both finance and AI research fields typically either examine enterprises' internal financial aspects to detect intra-risk (i.e., risks resulting from enterprises' operations), or they analyze risk

* Corresponding authors.
E-mail addresses: zhaoyu@swufe.edu.cn (Y. Zhao), kougang@swufe.edu.cn (G. Kou).

[1] https://www.worldbank.org/en/topic/smefinance [Accessed on 2023-10-22].

https://doi.org/10.1016/j.ins.2023.120081
Received 17 May 2023; Received in revised form 15 November 2023; Accepted 28 December 2023

Available online 4 January 2024

Table 1
The statistical significance analysis on the correlation between the enterprises' basic intelligence (including the enterprise basic attributes, the enterprise litigation information here) and their bankruptcy risk. The symbols ***, ** and * denote the statistical result is significant in the 99%, 95% and 90% confidence levels, respectively.

Enterprises intelligence	Significant indices	Correlation analysis		Independent Samples t-Test		
		Coefficient	Polarity	Average number of surviving enterprises	Average number of bankrupted enterprises	Significance value of average difference
Enterprise Attributes	Established time	-.058***	Negative	156	148	.000***
	Registered capital	-.187***	Negative	16874	910	.016**
	Paid-in capital	-.159***	Negative	16264	873	.020**
Lawsuit Cause	Loan contract dispute	.122***	Positive	1.80	2.23	.032**
	Sales contract dispute	.077***	Positive	.55	.80	.000***
Court Level of Lawsuit	Grassroots people's court	.086***	Positive	2.79	3.43	.019**
	Intermediate people's court	-.029**	Negative	.59	.43	.012**
	Higher people's court	-.070***	Negative	.05	.01	.000***
Verdict	Plaintiff winner	-.076***	Negative	.88	.24	.000***
	Defendant loser	.124***	Positive	1.87	3.12	.000***
Duration Of Action	Less than two years	.079***	Positive	3.23	3.82	.059*
	More than two years	-.086***	Negative	.19	.06	.000***

diffusion based on simulations [1] to mine contagion risk (i.e., risk from external stakeholders), including upstream and downstream companies and related persons. For simplicity, however, most studies only consider either **intra-risk** or **contagion risk** individually, ignoring their joint effect and thus fail to sufficiently model the enterprise risk and achieve state of art performances on downstream tasks. Given the heterogeneous multi-source characteristics of intra-risk data and the complexities of contagion risk relations among enterprises, building a framework for enterprise bankruptcy prediction that considers both intra-risk and contagion risk is a nontrivial and challenging task.

To meet this challenge, we propose a novel enterprise bankruptcy prediction method that combines intra-risk with contagion risk. First, we propose an intra-risk encoder that leverages rich features based on enterprises' basic business information and litigation information to mine intra-risk. After statistical analysis of the correlations between enterprises' basic intelligence (including basic business attributes and litigation information) and their bankruptcy risk (see Table 1), we select 12 statistically significant indices for intra-risk encoder learning (see Section 2.1 for details of the analysis of statistical significance).

Second, we propose an enterprise contagion risk encoder based on enterprise relational information from an enterprise knowledge graph (EKG) to embed contagion risk (also known as "risk momentum spillover effect" [2]). Fig. 1 shows a toy example of an EKG, from which we can find that enterprises have two kinds of relations: hyperedges and pair-wise heterogeneous relations (see Section 2.2 for details). Hence, we equip the contagion risk encoder with two submodels—hypergraph neural networks (HHGNN) and Hierarchical Graph Transformer Networks (HGTN)—to model risk diffusion in the EKG. Specifically, HHGNN aims to mine hyperedges in the EKG, such as the same industry and the same area, which is beneficial for enterprise risk prediction. During COVID-19, for example, most mask and vaccine manufacturers in the medical industry experienced a boom while the catering industry faced a significant bankruptcy risk. HGTN, meanwhile, can capture the direct contagion risk factors from neighboring enterprises. For example, an enterprise faces a loan default, which could lead to a bad financial situation for its creditors and potentially cause bankruptcy. Based on these two encoders, we propose a uniform framework for capturing both enterprise intra-risk and contagion risk for bankruptcy prediction. Fig. 2 shows the overall architecture of the proposed method.

In Fintech literature, especially in SMEs research, few researchers make their experimental benchmark datasets publicly available for reproduction.[2] This phenomenon, which could be related to the sensitivity of SMEs' financial data, results in a data deficiency that impedes SMEs' intelligence research [5]. In this study, we collect multi-sources SMEs data and build a new dataset (SMEsD), which we make publicly available. We hope the SMEsD will become a significant benchmark dataset for SMEs' bankruptcy prediction, and boost the development of financial risk research, especially SMEs bankruptcy research. Experimental results using the SMEsD demonstrate that our proposed model can sufficiently capture both intra-risk and contagion risk for bankruptcy prediction.

The contributions of this work are fourfold:

- We conduct exploratory data analysis to demonstrate that the enterprise intelligence (i.e., enterprise basic attributes, litigation information and the EKG) affects bankruptcy risk prediction for SMEs.
- We propose a novel framework for inferring enterprise bankruptcy by considering both intra-risk and contagion risk. To the best of our knowledge, this is the first attempt to consider both risks simultaneously and their joint effects in bankruptcy prediction.

[2] There are several top conference papers that did not provide their datasets for reproduction. Examples include SemiGNN [3] at ICDM 2019; HACUD [4] at AAAI 2019; ST-GNN [5] at IJCAI 2020; AMG-DP [6] at CIKM 2020; TemGNN [7] at SDM 2021; PC-GNN [8] at WWW 2021.

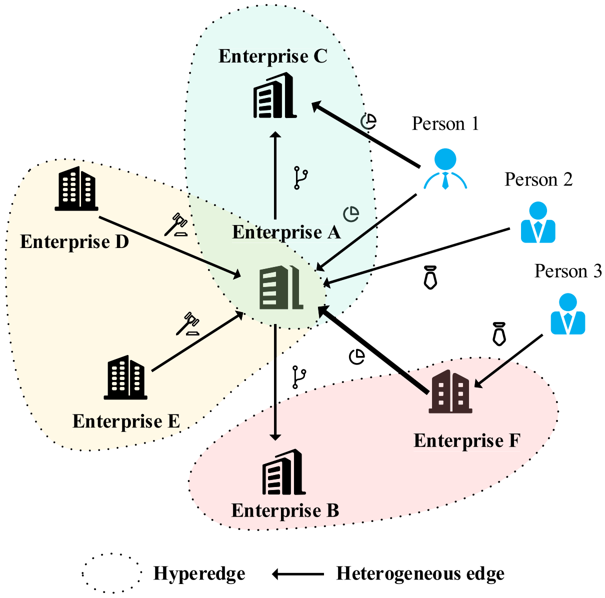

Fig. 1. A toy example of enterprise knowledge graph which is extracted from the newly constructed dataset SMEsD.

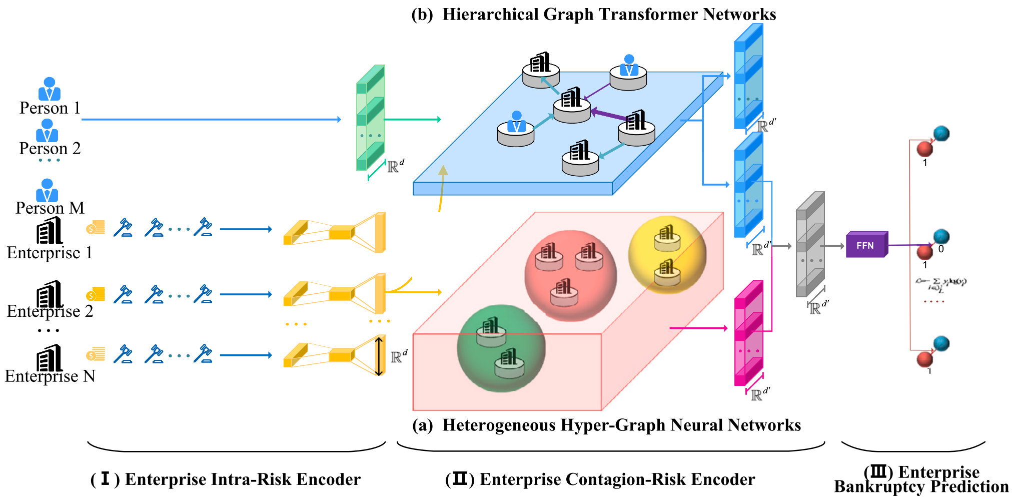

Fig. 2. The overall architecture of the proposed method. **(I) Enterprise Intra-Risk Encoder** using the enterprise statistically significant features in Table 1. **(II) Enterprise Contagion-Risk Encoder** is equipped with two sub-models: **(a) Heterogeneous Hyper-Graph Neural Networks** using enterprise hypergraph, **(b) Hierarchical Graph Transformer Networks** using enterprise heterogeneous graph, and **(c) Combining intra- and contagion risk**. **(III) Enterprise Bankruptcy Prediction**.

- Under this framework, we utilize an intra-risk encoder to derive intra-risk from an enterprise's basic intelligence. We propose a novel GNNs based contagion risk encoder that includes HHGNN and HGTN to calculate contagion risk based on hyperedges and pair-wise heterogeneous relations in the EKG.
- We propose a new benchmark dataset (SMEsD) to evaluate the proposed method, which is also expected to further promote enterprise financial risk analysis. The empirical experiments using our dataset demonstrate that the proposed method can successfully combine enterprise intra-risk and contagion risk for bankruptcy prediction.[3]

In conclusion, this work for the first time takes both intra-risk and contagion risk into consideration for inferring enterprise bankruptcy. We propose an enterprise intra-risk encoder with statistically significant indices to infer enterprise intra-risk. To model the enterprise contagion risk, we propose HHGNN and HGTN to calculate contagion risk based on hyperedges and pair-wise heterogeneous relations in the EKG. With those subtle designs, this work is able to capture complex enterprise risk sufficiently.

[3] The codes and datasets for reproduction are released on GitHub: https://github.com/shaopengw/ComRisk.

2. Exploratory analysis

In this section, we conduct an exploratory analysis of the relationship between the enterprise intelligence (i.e., enterprise basic attributes and litigation information, and the EKG), and bankruptcy risk. We first give the results for the statistical correlations and Independent Samples t-Test results between the basic attributes and the lawsuit features of the enterprises and their bankruptcy status. Then, we introduce a contagion risk analysis of the EKG for bankruptcy prediction.

2.1. Statistical significance analysis

We collect 11,523 civil lawsuits for 3,976 Chinese SMEs from 2000 to 2021 and the basic attributes of these enterprises. Table 1 summarizes the statistical analysis of the correlations and the Independent Samples t-Test between the enterprises' basic intelligence (i.e., enterprise basic attributes and litigation information; see Definition 1) and their bankruptcy risk. The first part in Table 1 refers to enterprise basic attributes (i.e., established time, registered capital, and paid-in capital). The last four rows in Table 1 concern the most significant features of lawsuits (i.e., lawsuit cause, court level, verdict, and duration of action). We present the analysis results below.

Enterprise Attribute. The first part concerns enterprise basic attributes, including established time (counted by months), registered capital, and paid-in capital (counted by 10,000 yuan). From Table 1, we can find the following:

- All three indicators are significantly negatively correlated with bankruptcy. The indicators of surviving enterprises are significantly higher than those of bankrupted enterprises in the t Test.

This indicates that the longer the established time, the greater the registered capital, and the greater the paid-in capital, the lower the probability of bankruptcy.

Lawsuit Cause. We explore the correlation between lawsuit causes and enterprise bankruptcy. In Table 1, we find that both types of lawsuit causes (i.e., loan contract dispute and sales contract dispute) are significantly correlated with enterprise bankruptcy. Specifically, the correlation coefficient between the number of loan contract disputes and enterprise bankruptcy is 0.122, which is statistically significant at the 99% level. The correlation coefficient between sales contract disputes and enterprise bankruptcy is 0.077, which is also statistically significant at the 99% level. These findings confirm that bankrupted enterprises tend to have more loan contract and sales contract disputes, which is in line with intuition. Meanwhile, we can observe that the average number of loan contract disputes among the surviving enterprises is 1.80, and the number for bankrupted enterprises is 2.23. The difference between the two is significant at the 95% level based on the t Test, which reaffirms the correlation between enterprise bankruptcy and loan contract disputes. We can obtain a similar conclusion from the statistical results for sales contract disputes. In summary, we find the following:

- The number of loan contract disputes and the number of sales contract disputes are both significantly positively correlated with enterprise bankruptcy.

Court Level of Lawsuit. The court level of a lawsuit is another factor related to enterprise risk. There are four levels of court types (from low to high): *grassroots people's court*, *intermediate people's court*, *higher people's court* and *supreme people's court*. Most lawsuits are dealt with by the grassroots people's court while some involving large underlying assets are brought to intermediate court. If the litigant disagrees with the verdict, it can appeal to a higher court. From Table 1, we can find the following:

- The number of grassroots court lawsuits is significantly positively correlated with enterprise bankruptcy.
- The lawsuit numbers of both intermediate people's court and higher people's court are significantly negatively correlated with enterprise bankruptcy.

These findings indicate that bankrupted enterprises tend to have more grassroots court lawsuits and fewer intermediate and higher court lawsuits. This could mean that being involved in a large number of grassroots court lawsuits implies that an enterprise has financial risk. Meanwhile, involvement in many high court lawsuits may reflect an enterprise's powerful capacity to deal with lawsuits, as well as its larger business scale. The t Test confirm this conclusion.

Verdict. We divide the results of lawsuits into four types according to litigant status and the verdict: plaintiff winner, plaintiff loser, defendant winner, and defendant loser. From Table 1, we can observe the following:

- Enterprises that are plaintiff winners are less likely to go bankrupt (i.e., significantly negative correlation).
- Enterprises that are defendant losers are more prone to bankruptcy (i.e., significantly positive correlation).

The correlation coefficients of the two types of verdicts are both significant at the 99% level, which confirms the importance of lawsuit results. The reason is that being a plaintiff winner in a lawsuit is good news for an enterprise, and being a defendant indicates risk. We can draw the same conclusion from the difference between the average number of the two types of lawsuit results for bankrupt and surviving enterprises in the t-Test.

Duration of Action. Referring to [9], we divide duration of action (DOA) into two types: less than two years and more than two years. From Table 1, we can find the following:

- The correlation between the number of lawsuits in the last two years and enterprise bankruptcy is significantly positive.
- The correlation between the number of lawsuits more than two years ago and enterprise bankruptcy is significantly negative.

These findings indicate that bankrupted enterprises tend to have more lawsuits in the two years prior to bankruptcy. The more lawsuits, the greater the direct risk for an enterprise, especially in the case of lawsuits in the last two years. Meanwhile, having been involved in a large number of lawsuits more than two years ago implies that an enterprise has experienced many disputes but has survived. This indicates that the enterprise has a large business scale and is strong enough to face various challenges.

2.2. Contagion-risk analysis

The contagion effect has been used to study stock movement prediction [10], in which stock fluctuations are partly affected by related stocks. In this study, contagion risk means that the risk generated by an enterprise tends to diffuse through the EKG to neighboring enterprises, which is ubiquitous in real market circumstances [5]. Fig. 1 shows an example of the EKG extracted from our newly generated dataset (SMEsD), from which we can find that enterprises have two types of relations: hyperedges and pair-wise heterogeneous relations. **(i)** There are three types of hyperedges in the EKG (see Definition 2): industry, area, and stakeholder, colored red, yellow, and green, respectively. For example, enterprise A, enterprise D, and enterprise E are in the same city. Then, they are influenced by the same regional policies (e.g., tax administration and economic policy) and face similar regional risks. Hence, we propose a Hyper-GNN to model such contagion risk. **(ii)** There are five types of pair-wise heterogeneous relations among enterprises and persons (see Definition 3). Person 1 and enterprise F both invest in enterprise A, where the edge widths indicate distinct investment share. Person 2 and person 3 are stakeholders (e.g., manager, stockholder, or supervisor) in enterprise A and enterprise F, respectively. Here, we use HGTN to model this type of contagion risk.

3. Problem formulation

In this section, we formally define the main terminologies and the problem, including enterprise basic intelligence, enterprise hyper-graph, enterprise heterogeneous-graph and enterprise bankruptcy prediction.

Definition 1. Enterprise basic intelligence. Enterprise basic intelligence consists of two parts (i.e., the enterprise basic attributes and the enterprise litigation information as is shown in Table 1), which can be formulated as $\mathcal{A} = (\mathcal{B}, \mathcal{J})$. Here, $\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_i, ..., \mathbf{b}_N\}$ denotes a set of enterprise basic business information. N denotes the number of enterprises. $\mathbf{b}_i = (et_i, rc_i, pc_i)$ denotes attributes for enterprise i, including established time, registered capital and paid-in capital. $\mathcal{J} = \{\mathcal{J}_1, \mathcal{J}_2, ..., \mathcal{J}_i, ..., \mathcal{J}_N\}$ denotes enterprise litigation information of enterprises. $\mathcal{J}_i = \{\mathbf{j}_i^1, \mathbf{j}_i^2, ..., \mathbf{j}_i^k, ..., \mathbf{j}_i^K\}$ denotes lawsuit set for enterprise i and K denotes the number of lawsuits of enterprise i. Besides, $\mathbf{j}_i^k = (lc_i^k, cl_i^k, vt_i^k, \Delta_i^k)$ denotes a specific lawsuit k related to enterprise i, including lawsuit cause, court level of lawsuit, verdict and time interval of action.

Definition 2. Enterprise hyper-graph. An enterprise hypergraph can be defined as $\mathcal{G}_{hyper} = (\mathcal{V}_e, \mathcal{E}, \mathcal{T}_{hyper})$. Here, $\mathcal{V}_e$ denotes the set of enterprise nodes. $\mathcal{E} = \{hp_1, hp_2, ...\}$ denotes hyperedge set. $\mathcal{T}_{hyper} = \{\Omega_1, \Omega_2, ..., \Omega_M\}$ denotes hyperedge type set, and $|\mathcal{T}_{hyper}| > 1$ here. Hyperedge type map function ψ: $\psi(hp) \in \mathcal{T}_{hyper}$. The relationship between enterprise nodes can be represented by an incidence matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$ with elements defined as:

$$\mathbf{H}(v, hp) = \begin{cases} 1, & \text{if } v \in hp \\ 0, & \text{otherwise} . \end{cases} \tag{1}$$

$v \in \mathcal{V}_e$ denotes an enterprise node, and $hp \in \mathcal{E}$ denotes a hyperedge.

Definition 3. Enterprise heterogeneous-graph. An enterprise heterogeneous graph is defined as a connected graph $\mathcal{G}_{hete} = (\mathcal{V}, \mathcal{L}, \mathcal{T}, \mathcal{R}, \mathbf{W})$. Here, $\mathcal{V}$ denotes the set of all nodes and $\mathcal{L}$ denotes the link set. They are associated with two functions: (i) a node type mapping function $\varphi : \mathcal{V} \to \mathcal{T}$, where $\mathcal{V} = \mathcal{V}_e \cup \mathcal{V}_p$. Specifically, $\mathcal{V}_e$ and $\mathcal{V}_p$ denote the node set of enterprises and persons, respectively. $\mathcal{V}_e \cap \mathcal{V}_p = \emptyset$. Each node $v \in \mathcal{V}$ belongs to one particular type in node type set $\mathcal{T} : \phi(v) \in \mathcal{T}$. (ii) a link class mapping function $\psi : \mathcal{L} \to \mathcal{R}$. And $\mathbf{W}$ denotes edge weights.

Problem 1. Enterprise bankruptcy prediction. Given an enterprise multi-source data, which consists of enterprise basic intelligence $\mathcal{A}$, an enterprise heterogeneous hypergraph $\mathcal{G}_{hyper}$ and an enterprise heterogeneous graph $\mathcal{G}_{hete}$, we aim to determine enterprise risk, considering both intra-risk and contagion risk. Based on enterprises' representations, we conduct bankruptcy prediction task, which can be treated as a binary classification problem.

4. Related work

In this section, we survey related work on two main aspects: enterprise risk analysis (4.1) and graph neural networks (4.2). The former includes current study on enterprise intra-risk and enterprise contagion-risk. The latter consists of studies of GNNs methods on enterprise risk and hypergraph neural networks.

4.1. Enterprise risk analysis

Enterprise Intra-Risk. In general, traditional enterprise risk analysis methods mainly consider financial indicators, such as profitability, operating efficiency, and solvency, using multivariate discriminant analyses [11] or machine learning methods [12], such as SVM and decision trees [13]. For example, Erdogan et al. [14] propose an ensemble method utilizing SVMs as base classifiers for commercial bank bankruptcy. Many other studies use neural networks to improve prediction accuracy. For example, Hosaka et al. [15] transform financial ratios into images using convolutional networks for bankruptcy prediction. Recently, many studies have focused on using text information, such as financial reports and conference calls, to mine enterprise intra-risk. For instance, Borochin et al. [16] find that the tone of conference calls is negatively related to firm value uncertainty in the equity options market. Li et al. [17] develop a large-scale multi-modal dataset called MAEC, and their experiments demonstrate the efficiency of the dataset for volatility forecasting. Cai et al. [18] utilize TextCNN to capture companies' related text semantic information for risk identification.

However, SMEs usually lack normal financial reports as well as public conference calls, which poses challenges for the analysis of SMEs. On the other hand, there are abundant risk sources such as relevant lawsuits, which are known to be significantly related to enterprise credit risk [9], which have not been well utilized in previous works.

Enterprise Contagion-Risk. Enterprise contagion-risk is also an important part of risk analysis since no enterprise is completely independent of other companies. Some financial studies propose using interconnections between firms or assets for risk analysis. Elsinger et al. [19], for example, propose assessing systemic financial stability using a network model of interbank loans. Acemoglu et al. [20] provide a framework for studying the relationship between financial network architecture and the likelihood of systemic failure considering contagion risk and find that financial contagion exhibits a form of phase transition as interbank connections increase. Kou et al. [21] also confirm that payment network-based information is critical for predicting bankruptcy in SMEs. Xie et al. [22] use a game theoretical approach to estimate contagion risk.

Most previous studies, however, explore the effects of contagion risk using simulations [1,23], which cannot be applied to real scenarios.

4.2. Graph neural networks

Graph neural networks (GNNs) use deep neural networks to deal with graph representation learning. They have proven to be successful for various tasks on graphs, such as node classification [24], link prediction [25] and graph classification [26]. GNNs also contribute to traditional scenarios, such as recommendation systems [27], natural language processing [28], and computer vision [29]. See [30] for more GNN surveys.

The interconnections between enterprises naturally form a heterogeneous graph, consisting of the enterprise nodes, the person nodes, and the connections among them. In the fintech field, some studies use GNNs to model various risks. For example, SemiGNN [3] involves using labeled and unlabeled multiview data for fraud detection. Hu et al. [6] model various relations, objects, the rich attributes of nodes, and edges for loan default detection. Cheng et al. [31] consider contagion to rate the risk of m guarantee chains in the bank industry. They employ a novel temporal inter-chain attention network to calculate the risk scores for contagion chains, which achieves significant improvement over baselines. Yang et al. [5] examine supply chain relationships and conduct lift prediction on a collected supply chain dataset. Zheng et al. [32] used a triple-layer attention network for bankruptcy prediction considering different metapath-based neighbors. Bi et al. [33] leverage shareholder information and financial news to construct a tribe-style graph and model the network with two levels for risk assessment. Different from previous work, Cheng et al. [34] propose to find critical companies to stem contagion risk in the bank industry with graph-based deep reinforcement learning.

Hypergraphs have shown a strong capacity to model higher-order relationships, which have been used in many areas, such as graph classification [35] and computer vision [36]. For instance, Zeng et al. [36] propose a hyperlink-based solution, MKHG, to bridge heterogeneous sources and heterogeneous modalities. With regard to enterprise risk modeling, there is a large number of hyperedges among enterprises and related persons, which is suitable for using hypergraphs. Few studies, however, have applied hypergraph neural networks in this area.

As discussed previously, few researchers have considered both intra-risk and contagion risk simultaneously in relation to bankruptcy prediction. Further, most fail to sufficiently mine risk information because of complex risk sources and relationships. Meanwhile, few studies provide open access data for other researchers, which restricts the development of risk analysis research in areas such as bankruptcy prediction and default prediction.

5. Methodology

In this section, we introduce the overall architecture of the proposed method, as shown in Fig. 2. The proposed model consists of three main parts: **(I)** the enterprise intrarisk encoder, which uses statistically significant enterprise features (Table 1); **(II)** Enterprise Contagion Risk Encoder, which consists of two submodules: (a) HHGNN, using enterprise hypergraphs, and (b) HGTN,

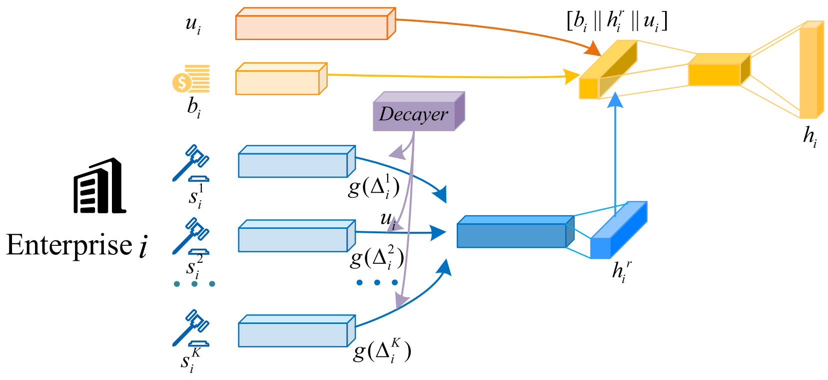

Fig. 3. Enterprise Intra-Risk Encoder.

using enterprise heterogeneous graphs; and (c) Combining intra- and contagion risk. **(III)** Enterprise Bankruptcy Prediction. Different from previous work, we take advantage of a hierarchical mechanism for both HHGNN and HGTN to utilize complex heterogeneous hyperedges and relationships. We provide the details below.

5.1. *Enterprise intra-risk encoder*

The enterprise intra-risk encoder aims to learn enterprise self-risk embedding using enterprise basic intelligence (i.e., enterprise basic attributes and enterprise litigation information), which is formally given in Definition 1.

As Fig. 3 shows, first, for each enterprise node $v_i \in \mathcal{V}_e$, we use $\mathbf{b}_i \in \mathbb{R}^{\hat{d}}$ in Definition 1 as the basic attribute features. Specifically, each enterprise possesses $\hat{d}$ basic attributes, such as established time, registered capital and paid-in capital, which form a $\hat{d}$-dimension vector in a $\hat{d}$-dimension vector space $\mathbb{R}^{\hat{d}}$. Second, the lawsuit event $\mathbf{j}_i^k$ of enterprise i contains four significance attributes (i.e., lawsuit cause, court level, verdict, and DOA), as described in Section 2.1. For the first three attributes, we map each into latent spaces and then concatenate them to obtain lawsuit representation $\mathbf{s}_i^k \in \mathbb{R}^{\tilde{d}}$.

And $\mathbb{R}^{\tilde{d}}$ denotes the transformed vector space, where the length of each vector is $\tilde{d}$. Referring to [37], we use a time decay function *Decayer* to weigh each lawsuit representation to make better use of time information in lawsuit events. Specifically, we calculate the time interval Δ_i^k between the time of occurrence of each lawsuit and the enterprise's observation time. For bankrupted enterprises, the observation time is set as the time of bankruptcy while for surviving enterprises, it is set as the present.

$$g(\Delta_i^k) = \frac{1}{1 + w \cdot \Delta_i^k} . \tag{2}$$

Because lawsuits in the past two years play an important role in enterprise risk prediction [9], we assign a lower w when performing time weight decay for lawsuits in the last two years.

Then, we aggregate lawsuit information from different time periods as follows:

$$\mathbf{h}_i^r = \sum_{k \in K_i} \mathbf{W}_{risk}\, g(\Delta_i^k) \cdot \mathbf{s}_i^k , \tag{3}$$

where $\mathbf{W}_{risk} \in \mathbb{R}^{\tilde{d} \times d}$ is a trainable matrix, $\mathbf{h}_i^r$ is the aggregated lawsuit information of company i.

We also generate a pre-trained embedding $\mathbf{u}_i \in \mathbb{R}^{\overline{d}}$ for enterprise i as a supplement embedding. Finally, we concatenate the basic attribution features, litigation embedding, and supplement embedding and project it into a new latent space as follows:

$$\mathbf{h}_i = \mathbf{W}_e \cdot [\mathbf{b}_i || \mathbf{h}_i^r || \mathbf{u}_i] , \tag{4}$$

where $\mathbf{h}_i$ denotes the output of intra-risk representation of the enterprise i and $||$ denotes the concatenation operation. $\mathbf{W}_e \in \mathbb{R}^{(\hat{d}+d+\overline{d}) \times d}$ is a trainable matrix.

5.2. *Enterprise contagion risk encoder*

5.2.1. *Heterogeneous Hyper-Graph Neural Networks*

Hypergraphs play an important role in bankruptcy prediction, as the hyperedges reflect common factors that enterprises face. Thus, it is natural to utilize hypergraphs to capture common risk information, such as industry development recession, regional economic policy changes, and guarantee risk caused by the same stakeholders.

As shown in Fig. 4, because different types of hyperedges contribute to node representation at different levels, we assign different weights to them when aggregating node representations. Specifically, following Feng et al. [38], we first calculate the hypergraph convolution module as follows:

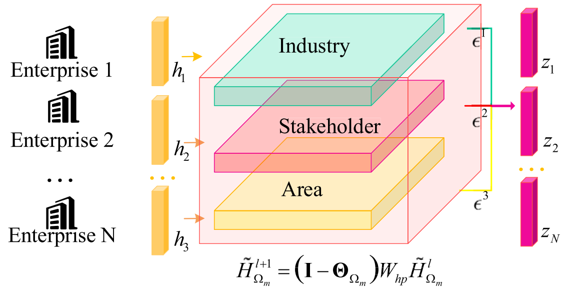

Fig. 4. Heterogeneous Hyper-Graph Neural Networks.

$$\underbrace{\begin{array}{c|ccc} & \Omega_m^{\;1} & \Omega_m^{\;2} & \Omega_m^{\;3} \\ \hline v_1 & 1 & 1 & 0 \\ v_2 & 1 & 0 & 1 \\ v_3 & 1 & 1 & 0 \\ v_4 & 0 & 1 & 1 \end{array}}_{H_{\Omega_m}} \times \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{W} = \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}}_{H_{\Omega_m} W}$$

Fig. 5. A toy example of the hyperedge weight matrix **W**.

$$\mathbf{\Theta}_{\Omega_m} = \mathbf{D}_v^{-1/2} \mathbf{H}_{\Omega_m} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}_{\Omega_m}^{\top} \mathbf{D}_v^{-1/2} , \tag{5}$$

where $\mathbf{\Theta}_{\Omega_m} \in \mathbb{R}^{|\mathcal{V}_e| \times |\mathcal{V}_e|}$ denotes the convolution module. $\mathbf{D}_v$ is the enterprise node degree matrix. $\mathbf{H}_{\Omega_m}$ denotes the incident matrix of the hypergraph type Ω_m, and $\mathbf{D}_e$ denotes the hyperedge degree matrix.

Note that $\mathbf{W}$ represents the hyperedge weight matrix. We've defined it as an identity matrix, indicating that all weights are equal. As illustrated in Fig. 5, suppose we have four nodes and three hyperedges within the hypergraph type Ω_m. From this, it becomes evident that nodes v_1, v_2, and v_3 are linked by the same hyperedge, Ω_m^1, and that node v_1 belongs to both Ω_m^1 and Ω_m^2. Furthermore, it's essential to note that when $\mathbf{W}$ is set as an identity matrix, we have $\mathbf{H}_{\Omega_m}\mathbf{W} = \mathbf{H}_{\Omega_m}$, signifying that no single hyperedge holds greater or less significance within the scope of this computation.

Afterwards, we conduct hypergraph convolution under the hypergraph type Ω_m as follows:

$$\widetilde{\mathbf{H}}^{l+1}_{\Omega_m} = \left(\boldsymbol{I} - \mathbf{\Theta}_{\Omega_m} \right) \mathbf{W}_{hp} \widetilde{\mathbf{H}}^{l}_{\Omega_m}, \tag{6}$$

where $\widetilde{\mathbf{H}}^{l+1}_{\Omega_m}$ denotes the learned representations under the hypergraph type Ω_m of layer $l+1$. Besides, $\boldsymbol{I} - \mathbf{\Theta}_{\Omega_m}$ denotes the hypergraph laplacian and $\mathbf{W}_{hp} \in \mathbb{R}^{\mathbf{d} \times \mathbf{d}'}$ is a trainable matrix, which is shared for different types of hypergraphs. Then we aggregate the different types of hypergraph convolution representations as follows:

$$\mathbf{z}_i = \sum_{\Omega_m \in \mathcal{T}_{hyper}} \epsilon^{\Omega_m} \cdot \widetilde{\mathbf{h}}_i^{\Omega_m}, \tag{7}$$

where $\mathbf{z}_i \in \mathbb{R}^{d'}$ is the learned hypergraph comprehensive representation of enterprise i, and ϵ^{Ω_m} is a trainable parameter, which denotes the importance of hypergraph Ω_m for all enterprise nodes.

5.2.2. Hierarchical Graph Transformer Networks

We propose the HGTN to sufficiently make use of multiplex interactions among enterprises and persons. Specifically, we first aggregate entity level information and then relationship level in a hierarchical mechanism as shown in Fig. 6.

We initialize the person node representations the same as for enterprises in Section 5.1. Then we perform transformation based on node type to project enterprise node and person representation to same latent space as follows:

$$\mathbf{h}_i' = \mathrm{Norm}(\mathbf{W}_{\phi(v_i)} \mathbf{h}_i) , \tag{8}$$

where $\mathbf{W}_{\phi(v_i)} \in \mathbb{R}^{d \times d'}$ is a node type specific trainable weight matrix. $\mathbf{h}_i \in \mathbb{R}^d$ and $\mathbf{h}_i' \in \mathbb{R}^{d'}$ are the original and transformed node representations, respectively. Norm denotes Batch Normalization operation [39]. Then we conduct entity level aggregation. For weighted edges, such as *holder_investor*, we directly set the ratio of contribution capital as the edge weight. For unweighted relations, we use the attention mechanism to assign weights for node v_i's neighbors' representation as follows:

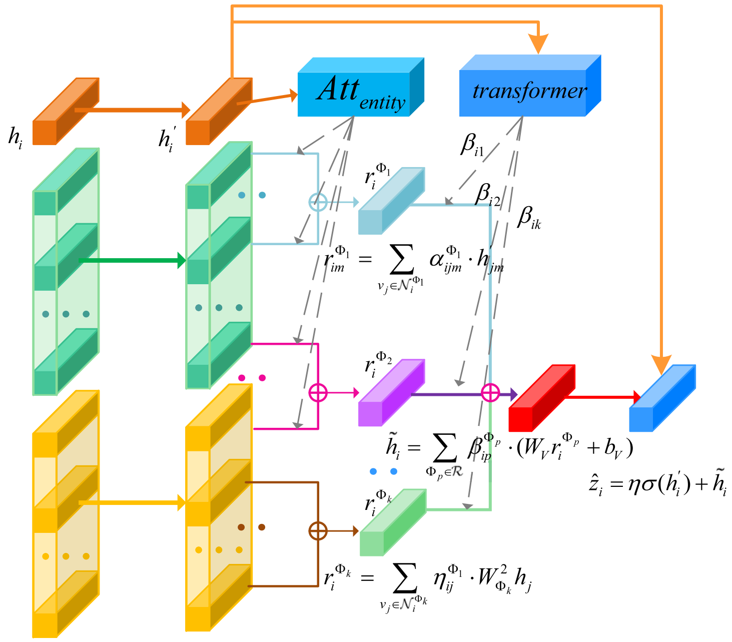

Fig. 6. Hierarchical Graph Transformer Networks.

$$
\begin{aligned}
\mathbf{e}_{ij}^{\Phi_k} &= \mathrm{Att}_{entity}(\mathbf{h}_i', \mathbf{h}_j'; \Phi_k) \\
&= \mathrm{LeakyRelu}(\mathbf{W}_{\Phi_k}^1 \cdot [\mathbf{h}_i' \| \mathbf{h}_j']) ,
\end{aligned} \tag{9}
$$

where $\mathbf{e}_{ij}^{\Phi_k}$ is the learned importance of node i's neighbor j under relationship Φ_k. Note that $\mathbf{W}_{\Phi_k}^1 \in \mathbb{R}^{2d' \times d'}$ is a trainable matrix, and LeakyRelu is an activation function. To make the weights comparable, we utilize *Softmax* function to normalize weights across all choices of j as follows:

$$
\alpha_{ijm}^{\Phi_k} = Softmax_j(e_{ijm}^{\Phi_k}) = \frac{\exp(e_{ijm}^{\Phi_k})}{\sum_{v_p \in \mathcal{N}^{\Phi_k}(v_i)} \exp(e_{ipm}^{\Phi_k})} , \tag{10}
$$

$$
r_{im}^{\Phi_k} = \sum_{v_j \in \mathcal{N}_i^{\Phi_k}} \alpha_{ijm}^{\Phi_k} \cdot h_{jm}' , \tag{11}
$$

where $r_{im}^{\Phi_k}$ is the m-th element of the aggregated Φ_k unweighted relationship representation for node v_i. Note that $\alpha_{ijm}^{\Phi_k}$ is the m-th dimension of the normalized importance of node j related to node i under the unweighted relationship Φ_k, and $\mathcal{N}_i^{\Phi_k}$ denotes node i's neighbors under unweighted relationship Φ_k. For weighted edges, we implement node level aggregation as follows:

$$
\eta_{ij}^{\Phi_k} = Softmax_j(w_{ij}^{\Phi_k}) = \frac{\exp(w_{ij}^{\Phi_k})}{\sum_{v_p \in \mathcal{N}^{\Phi_k}(v_i)} \exp(w_{ip}^{\Phi_k})} , \tag{12}
$$

$$
\mathbf{r}_i^{\Phi_k} = \sum_{v_j \in \mathcal{N}_i^{\Phi_k}} \eta_{ij}^{\Phi_k} \cdot \mathbf{W}_{\Phi_k}^2 \mathbf{h}_j , \tag{13}
$$

where $\eta_{ij}^{\Phi_k}$ denotes the normalized importance that node j has for node i under weighted relation, and $w_{ij}^{\Phi_k}$ denotes original edge weight between node i and node j (e.g., such as contribution capital). $\mathbf{W}_{\Phi_k}^2 \in \mathbb{R}^{d' \times d'}$ is a trainable matrix. $\mathbf{r}_i^{\Phi_k}$ denotes the learned aggregated representation of node i's neighbors under the weighted relationship Φ_k.

To fully capture the risk information implied in different relationships, we use transformer based attention mechanism:

$$
\begin{aligned}
g_{ik} &= \mathbf{k}_i^{\top} \mathbf{q}_i \cdot \frac{\mu_{\Phi_k}}{\sqrt{d'}}, \\
\mathbf{q}_i &= \mathbf{W}_Q^{\Phi_k} \mathbf{h}_i' + \mathbf{b}_Q^{\Phi_k} , \\
\mathbf{k}_i &= \mathbf{W}_K^{\Phi_k} \mathbf{r}_i^{\Phi_k} + \mathbf{b}_K^{\Phi_k} ,
\end{aligned} \tag{14}
$$

where g_{ik} denotes the relationship level importance that relation Φ_k has for node i. Note that $\mathbf{W}_Q^{\Phi_k}, \mathbf{W}_K^{\Phi_k} \in \mathbb{R}^{d' \times d'}$ are trainable matrices in relationship Φ_k. Besides, $\mathbf{b}_Q^{\Phi_k}$ and $\mathbf{b}_K^{\Phi_k} \in \mathbb{R}^{d'}$ are trainable parameters in relationship Φ_k, and μ_{Φ_k} is a trainable parameter used to adjust the scale of learned importance, which is relationship type specific. Similarly, we utilize the *Softmax* function to normalize learned attention and aggregate relation level representations as follows:

$$\beta_{ik} = Softmax_k(g_{ik}) = \frac{\exp(g_{ik})}{\sum_{\Phi_p \in \mathcal{R}} \exp(g_{ip})} , \tag{15}$$

$$\widetilde{\mathbf{h}}_i = \sum_{\Phi_p \in \mathcal{R}} \beta_{ip}^{\Phi_p} \cdot (\mathbf{W}_V \mathbf{r}_i^{\Phi_p} + \mathbf{b}_V) , \tag{16}$$

where β_{ik} denotes the normalized importance of relationship Φ_k for node i, and $\mathbf{W}_V \in \mathbb{R}^{d' \times d'}$ and $b_V \in \times\mathbb{R}^{d'}$ are trainable parameters. $\widetilde{\mathbf{h}}_i$ is the learned aggregated risk information for node i. Next, we use the residual connection to get the final risk information of the heterogeneous graph as follows:

$$\hat{\mathbf{z}}_i = \eta\sigma(\mathbf{h}'_i) + \widetilde{\mathbf{h}}_i , \tag{17}$$

where η is the learned weight to balance the aggregated risk information and nodes' initial risk information, σ is the GELU [40] activation function and $\hat{\mathbf{z}}_i$ is the final risk information of node i.

5.2.3. Combining intra- and contagion risk

We sum the propagated risk from the HHGNN and HGTN to get the contagion-risk as follows:

$$\mathbf{z}_i^{cont} = \mathbf{W}^{cont} \cdot (\mathbf{z}_i + \hat{\mathbf{z}}_i) , \tag{18}$$

where $\mathbf{W}^{cont} \in \mathbb{R}^{d' \times d'}$ is a trainable matrix and $\mathbf{z}_i^{cont}$ is the learned contagion risk.

Then, we combine the node intra-risk and contagion-risk information as follows:

$$\bar{\mathbf{z}}_i = \lambda\sigma(\mathbf{z}_i^{cont}) + (1-\lambda)\mathrm{MLP}(\widetilde{\mathbf{h}}_i) , \tag{19}$$

where $\bar{\mathbf{z}}_i$ is the final representation of node i, and λ is a trainable parameter to balance contagion risk and intra-risk. σ is an activation function, we choose GELU here. MLP is a two-layer multilayer perception with the ReLU [41] activation function in it.

5.3. Optimization

We sum the learned representations of HHGNN and HGTN and utilize a fully connected layer to transform learned node representations for bankruptcy prediction, as in Fig. 2 (III).

$$\tilde{y}_i = Softmax\left(\mathbf{W}_p \bar{\mathbf{z}}_i + \mathbf{b}_p\right) , \tag{20}$$

where $\mathbf{W}_p$ is a trainable matrix and $\mathbf{b}_p$ is the bias vector. Finally we train the model by minimizing cross-entropy loss.

$$\mathcal{L} = -\sum_{i \in \mathcal{Y}_L} y_i \log(\tilde{y}_i) , \tag{21}$$

where $\mathcal{Y}_L$ is the set of labeled nodes. y_i and $\tilde{y}_i$ are the ground truth and the predicted bankruptcy probability for node i, respectively.

5.4. Discussion of ComRisk

To model the complex components of risk in the EKG, we propose *ComRisk*, a novel method that can take advantage of both enterprise intra-risk and contagion risk. We first utilize an enterprise intra-risk encoder to learn enterprise self-risk caused by their basic intelligence. To fully capture the contagion risk, we propose two critical sub-modules (i.e., *Heterogeneous Hyper-graph Neural Networks* and *Hierarchical Graph Transformer Networks*) to model common risk and direct risk, respectively. The former aims to model heterogeneous environment risk on hyper-graphs. The latter models complex relationships, which are the carriers of various direct contagion risks, taking advantage of both the hierarchical mechanism and transformer-style computing architecture. Equipped with the above design, the proposed model is able to capture comprehensive risk in the EKG and promote performances on downstream tasks.

6. Experiments

In this section, we illustrate the experiments, including experiment settings (6.1), experiment details (6.2), the main results and analysis (6.3), node clustering (6.4), ablation study (6.5), variant analysis (6.6), visualization (6.7) and parameter analysis (6.8).

Table 2
Statistics of the SMEsD.

SMEsD		Train	Validation	Testing
Node	*#company*	2848	741	505
	#person	1752	367	322
HeteG	*#manager*	2658	724	562
	#shareholder	4002	1016	704
	#other stakeholder	4426	1028	948
	#holder_investor	6626	1574	1208
	#branch	594	98	76
HyperG	*#industry*	108	82	68
	#area	153	61	61
	#stakeholder	756	164	152
Label	*#bankrupt*	1621	354	318
	#survive	1195	367	173

6.1. Experimental settings

6.1.1. Datasets

To examine the performance of the proposed model for bankruptcy prediction, we manually collected and preprocessed a real-world SME dataset, which we call **SMEsD**. Specifically, we chose bankrupted companies in Zhejiang province, one of China's most prosperous provinces. We started from the initial seed set, which included 889 companies, and found out their related companies and persons regarding shareholder relationships, investment relationships, and executives. We crawled all the public information from the Internet. To the best of our knowledge, this dataset is the largest multimode bankruptcy prediction dataset, compared with most previous research that is based on around 100 companies, and it contains abundant multidimensional information. SMEsD consists of 3,976 SMEs and related persons in China from 2014 to 2021, constituting a complex EKG. All enterprises are associated with their basic business information and lawsuit events spanning 2000–2021. Specifically, enterprise business information includes registered capital, paid-in capital, and established time. Each lawsuit consists of the associated plaintiff, defendant, subjects, court level, result, and timestamp.

Table 2 presents the statistics of the SMEsD. The dataset contains two types of nodes: enterprise and person. For the enterprise heterogeneous graph, there are five types of relationships between enterprises and persons. The *holder_investor* relationship is weighed by the contribution capital, and the other edges are unweighted. For the hypergraph, there are three types of edges: industry, area, and stakeholder. We split SMEsD into a training set, validation set, and testing set according to the bankruptcy time.

6.1.2. Baselines

To measure the effectiveness of our method, we compare the proposed model with four types of state-of-the-art (SOTA) methods: **(1)** the conventional machine learning based method that only considers enterprise lawsuit information including lawsuit attribute frequency and basic business information; **(2)** hypergraph neural networks based methods that take high-order relationships among enterprises into consideration; this can detect the common risks enterprises face; **(3)** homogeneous GNNs based methods that use abundant connections among enterprises, which can capture contagion risk; and **(4)** heterogeneous GNNs based methods that can distinguish complex relationships in an EKG.

Conventional Machine Learning (ML) Based Methods:

- Logistic Regression (**LR**) [42]: a well-known method applied in machine learning, social science and biometrics when explained variables are discrete.
- Support Vector Machine (**SVM**) [43]: a model that utilizes support vectors to divide vector spaces into different classes.
- Gradient Boosting Decision Tree (**GBDT**) [44]: a classic tree classification model of conventional machine learning.

Hypergraph Neural Networks (HyperG) Based Methods:

- Hypergraph Neural Networks (**HGNN**) [38]: a model proposed to utilize high-order relationship information in graphs.
- Hypergraph Wavelet Neural Network (**HWNN**) [35]: a newly proposed model that makes use of wavelet basis instead of Fourier basis to perform localized hypergraph convolution.

Homogeneous GNNs (HomoG) Based Methods:

- Graph Convolutional Networks (**GCN**) [24]: a popular model which averages neighbors' information during the message passing process.

Table 3
The overall performance.

Models		Accuracy	Precision	Recall	F1	AUC
ML	**LR (2013 [42])**	0.6090	0.6780	0.7547	0.7143	0.5812
	SVM (1999 [43])	0.6314	0.6612	0.8836	0.7564	0.5256
	GBDT (2001 [44])	0.6456	0.7449	0.6887	0.7157	0.6843
HomoG	**GCN (2017 [24])**	0.6619	0.6792	0.9057	0.7763	0.7099
	GAT (2018 [45])	0.6802	0.6998	0.8868	0.7822	0.6251
HyperG	**HGNN (2019 [38])**	0.6884	0.6941	**<u>0.9277</u>**	0.7941	0.6433
	HWNN (2021 [35])	0.6640	0.7029	0.8333	0.7626	0.6395
HeteG	**RGCN (2018 [46])**	0.6965	0.7464	0.8050	0.7746	0.6857
	HetGNN (2019 [47])	0.6965	0.7036	0.9182	0.7967	0.7185
	HAN (2019 [27])	<u>0.7332</u>	0.7429	0.8994	<u>0.8137</u>	0.7331
	ie-HGCN (2021 [25])	0.7210	<u>0.7521</u>	0.8491	0.7976	<u>0.7560</u>
	HAT (2021 [32])	0.7312	0.7435	0.8931	0.8114	0.7006
Loss-weighted ComRisk		0.7739	0.7820	0.9025	**0.8380**	0.8256
ComRisk		**0.7800**	**0.8409**	0.8145	0.8275	**0.8483**

- Graph Attention Networks (**GAT**) [45]: a recent model that takes attention mechanism to align different weights to neighbors during the information aggregating process.

Heterogeneous GNNs (HeteG) Based Methods:

- Relational Graph Convolutional Networks (**RGCN**) [46]: an advanced extension of GCN, which takes relationship information into consideration by giving different weights for different relationships.
- Heterogeneous Graph Neural Network (**HetGNN**) [47]: a multi-modal heterogeneous graph model that uses Bi-LSTM to process multi-modal information and then applies the attention mechanism in heterogeneous information fusing.
- Heterogeneous Graph Attention Network (**HAN**) [27]: one of the earliest models to implement hierarchical attention based on the metapath relationships in graph neural networks.
- interpretable and efficient Heterogeneous Graph Convolutional Network (**ie-HGCN**) [25]: a SOTA model that first implements object-level aggregation and then aggregates type-level information based on different metapaths.
- Heterogeneous-attention-network-based model (**HAT**) [32]: a SOTA model which conducts triple-level attention in SMEs bankrupt prediction.

6.2. Experiment details

For all of the baseline methods, we calculate enterprise risk information by counting the number of lawsuit attributes and combining them with basic enterprise business attributes as enterprise risk representations. We use random initialization based on standard normal distribution to assign initial representations for enterprises and persons when implementing GNN-based methods. We choose Metapath2vec [48] as the pre-trained model for ComRisk to generate the supplementary embeddings. We implement ComRisk and baselines with PyTorch and PyTorch Geometric (PyG). We refer to THU-HyperG [49] to construct the hypergraphs. We implement baselines based on official codes with fine-tuning parameters, including hidden dimension, layer number, and multihead number, to obtain better performance. All neural network–based models are trained with the Adam optimizer [50] and the Cosine Annealing Learning Rate Scheduler [51]. We set input dimension 16 and output dimension 12 for each model. We run all methods for 500 epochs and update the models considering the improvement of the two comprehensive indicators on the validation dataset (i.e., the accuracy and F1 score to alleviate the overfitting problem). We report the results of all methods on the testing dataset.

6.3. Experimental results and analysis

Table 3 shows the evaluation results against 12 SOTA baselines. We can see that the proposed method outperforms all baselines for enterprise bankruptcy prediction in terms of all of the comprehensive metrics on our newly generated dataset (SMEsD). Specifically, COMRISK achieves SOTA performance with improvements of 4.68%, 1.38%, and 9.23% for accuracy, F1, and AUC scores, respectively. This confirms the ability of our method to use both intrarisk and contagion risk for bankruptcy prediction.

Major Analysis. (1) We observe that the SVM achieves good performance for recall. This is because lawsuit information and basic enterprise information are highly correlated with enterprise bankruptcy. However, the SVM has poor performance on other comprehensive metrics because of overfitting. (2) We can observe that all graph based models, including hypergraph neural networks and Hierarchical Graph Transformer Networks, perform better on most metrics than machine learning methods. This demonstrates the superiority of using contagion risk for enterprise bankruptcy prediction. (3) We also find that HWNN performs better than HGNN because it considers different types of hyperedges; this demonstrates the necessity of considering hypergraph heterogeneity. (4) In

Table 4
Node clustering results.

Matrics	GCN	GAT	HWNN	HGNN	RGCN	HetGNN	HAN	ie-HGCN	HAT	ComRisk
ARI	0.0225	-0.0151	-0.0296	-0.0139	-0.0093	0.0346	-0.0045	0.0355	0.1407	**0.1639**
NMI	0.0712	0.0304	0.0156	0.0072	0.0104	0.0272	0.0035	0.1057	0.1509	**0.1724**

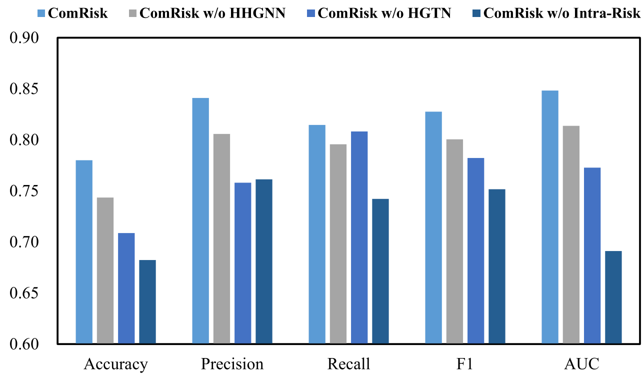

Fig. 7. Ablation study.

addition, we find that the two SOTA HeterG baseline models (i.e., ie-HGCN and HAN) show better performance than ML and HomoG models, which confirms the ability of heterogeneous graphs to capture contagion risk.

Credit Scenario Analysis. Enterprise bankruptcy prediction can be applied to the credit scenario. It is important to support SMEs in this regard since they contribute a great deal to the economy. In the past, banks preferred not to give loans to SMEs since the banks usually did not have reliable access to SMEs' risk levels, resulting in high recall scores in loan decisions. By contrast, precision is a better indicator that can help banks exclude high-risk enterprises and offer more loans to SMEs while avoiding losses. We can see in Table 3 that the proposed model ComRisk achieves an 8.88% gain over the SOTA baseline model, which can benefit both loan decision-makers and SMEs. Meanwhile, to promote the recall score, we propose Loss-weighted ComrRisk, which assigns more weight for bankrupt enterprises' losses during the training process. We can also see in Table 3 that Loss-weighted ComrRisk achieves comparable performance in recall and maintains excellent performance in other metrics at same time.

6.4. Node clustering

We conduct node clustering to validate the approach. Here, we first get enterprise node representations via feed-forward of each GNNs. We then apply K-Means to implement node clustering and evaluate the performances using NMI and ARI based on their ground truth and predicted categories. Since the results tend to be affected by initial centroids, to make the performances more stable, we repeat the former process 20 times and report average results in Table 4. Experimental results show that our model outperforms all baselines, e.g. ComRisk outperforms the SOTA model HAT with a margin as large as 0.0232 on ARI and 0.0215 on NMI, respectively. The results demonstrate the superiority of the learned node representations.

6.5. Ablation study

We conduct an ablation experiment to evaluate the effectiveness of different components in the proposed model ComRisk. The three ablated variants are as follows: (1) **ComRisk w/o Intra-Risk**, which deletes the inner risk encoder; (2) **ComRisk w/o HHGNN**, which removes the hierarchical hypergraph encoder; and (3) **ComRisk w/o HGTN**, which deletes the hierarchical risk encoder module. Fig. 7 shows the results. We can see that removing either the heterogeneous graph, hypergraph, or risk encoder leads to performance degeneration, which demonstrates the effectiveness of the three modules. Specifically, the proposed model ComRisk outperforms **ComRisk w/o Intra-Risk**, which confirms the effectiveness of lawsuit information for bankruptcy prediction. Meanwhile, **ComRisk w/o Intra-Risk** has the worst performance among the three ablated variants, which verifies the importance of intrarisk information. Thus, we highlight the design of capturing lawsuit risk information. Compared with **ComRisk w/o HHGNN**, the proposed model ComRisk also achieves better performance, which demonstrates the contribution of hypergraphs. This is because enterprises in the same industry, in the same area, or with same stakeholders usually face similar external risks (e.g., industry development recession, regional economic policy changes, and guarantee risks), which can be detected by hypergraphs. For **ComRisk w/o HGTN**, we find that performance also decreases, which confirms that utilizing complex heterogeneous relationships in an EKG can strengthen the capacity of the model.

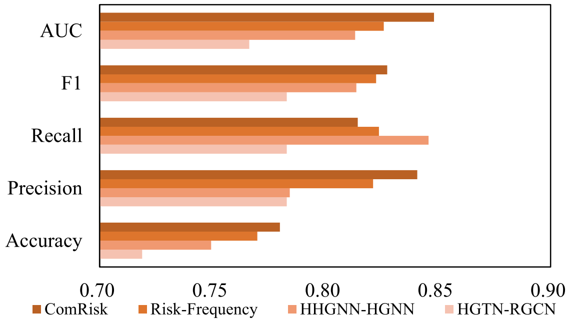

Fig. 8. Variants analysis.

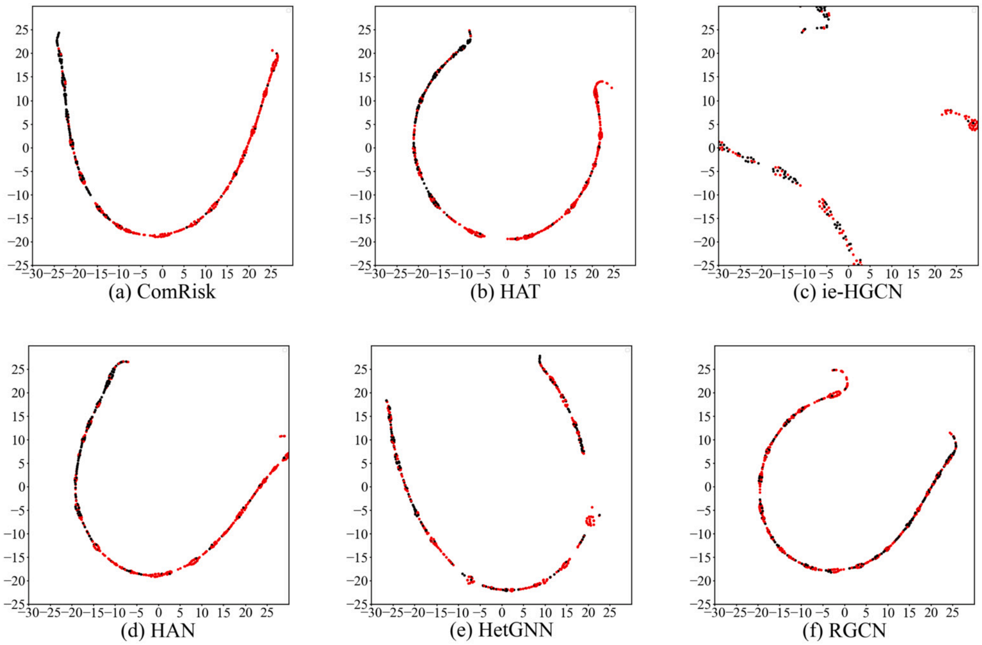

Fig. 9. Visualization of node representation. Red points denote survived enterprises, and black nodes denote bankrupted enterprises. Less mixed areas mean better performance. We can observe that the proposed ComRisk outperforms other models.

6.6. Variant analysis

We conduct a variant analysis of ComRisk to show the effectiveness of its architecture. (1) **ComRisk-Frequency** replaces the proposed inner risk encoder with the frequency of lawsuit attributes with regard to each enterprise; (2) **HHGNN-HGNN** replaces the hierarchical hypergraph encoder with HGNN; and (3) **HGTN-RGCN** uses RGCN rather than the proposed hierarchical risk encoder. Fig. 8 shows the results. We can see that the proposed ComRisk achieves the best performance compared to all variants. Specifically, ComRisk performs better than Risk-Frequency, which again demonstrates the risk-representation capacity of the inner risk encoder. This is because our model not only uses lawsuit risk information in terms of frequency, but also considers the time interval related to each lawsuit, which is shown to be significantly correlated with enterprise bankruptcy in Table 1. Compared with Hyper-HGNN, the proposed model ComRisk also performs better because it can distinguish different types of hyperedges and assign different importance weights for the learned representations. We also observe that replacing the hierarchical risk encoder with RGCN lowers performance, from which we can conclude that the proposed hierarchical risk encoder can better capture the contagion risk embedded in complex relationships.

6.7. Visualization

To make a more intuitive comparison, we project the representations of the enterprise nodes into a two-dimensional space using t-SNE [52]. The node representations are learned by the proposed ComRisk and heterogeneous graph neural networks. The color indicates the state of enterprises in Fig. 9. The less mixed areas, the better. We can observe that our model ComRisk performs best

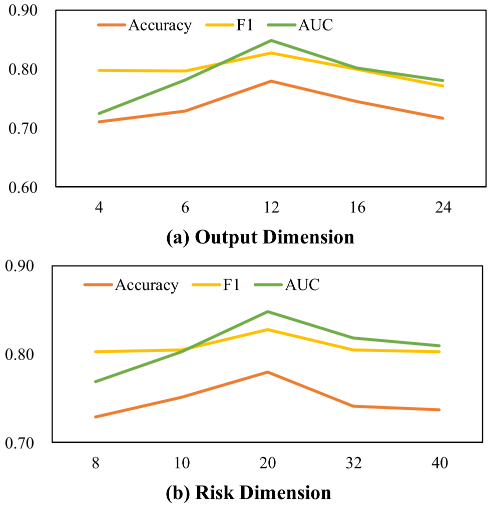

Fig. 10. Parameter analysis.

in visualization as there are more distinct boundaries and fewer mixed nodes. Besides, we also find that hierarchical-attention-based models (i.e., HAT and HAN) perform better than general heterogeneous graph models (i.e., ie-HGCN, HetGNN and RGCN).

6.8. Parameter analysis

We examine the effects of the two critical hyper-parameters (i.e., the output dimension and lawsuit risk information dimension in ComRisk), the default dimensions of which are 12 and 20, respectively.

Impact of input dimension. As shown in Fig. 10 (a), performance first increases with the dimension increasing before 12 and then falls with the dimension increasing. This could be because a model with a too-low dimension fails to represent abundant node information. Meanwhile, a high dimension produces too much noisy information and thus restricts the capacity of the model (ComRisk).

Impact of lawsuit risk information dimension. We can see in Fig. 10 (b) that model performance first increases and reaches its peak at 20 and then decreases with the dimension rising. This is mainly because the number of total lawsuit attributes in the SMEsD dataset is 20; lower and higher lawsuit risk dimensions both lead to a decrease in performance.

7. Conclusion

In this study, we propose modeling enterprise bankruptcy risk by combining intrarisk and contagion risk. In this framework, we propose a novel method that includes an intrarisk encoder and GNNs based contagion risk encoder. Specifically, the intrarisk encoder can capture enterprise intrarisk using statistically correlated indicators derived from basic business information and litigation information. The contagion risk encoder consists of hypergraph neural networks and heterogeneous graph neural networks, which aim to model contagion risk in the two aspects of hyperedge and complex heterogeneous relationships among EKGs, respectively. To evaluate the proposed model, we collect multisource SME data and build a new dataset, SMEsD. The experimental results demonstrate the superiority of the proposed method. The dataset is expected to become a significant benchmark dataset for SME bankruptcy prediction while further promoting research on financial risk. In the future, we will pay more attention to the modeling of large-scale and dynamic enterprise knowledge graphs, which bring more challenges but also contribute to the field. Besides, we will make efforts to provide explainable GNNs for enterprise risk mining. Finally, we will continue to provide more public multi-source enterprise knowledge graphs regarding different types of enterprise risk mining scenarios.

CRediT authorship contribution statement

Shaopeng Wei: Conceptualization, Formal analysis, Investigation, Methodology, Software, Writing – original draft. **Jia Lv:** Investigation. **Yu Guo:** Investigation. **Qing Yang:** Investigation. **Xingyan Chen:** Writing – review & editing. **Yu Zhao:** Funding acquisition, Supervision, Writing – review & editing. **Qing Li:** Writing – review & editing. **Fuzhen Zhuang:** Writing – review & editing. **Gang Kou:** Funding acquisition, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data and code can be found in the paper.

Acknowledgement

The research is supported by the Key Technologies Research and Development Program under Grant No. 2020YFC0832702 and National Natural Science Foundation of China under Grant Nos. 71910107002, 62376227, 61906159, 62176014, 71873108, 62072379, and Sichuan Science and Technology Program under Grant Nos. 2023NSFSC0032, 2023NSFSC0114, and Guanghua Talent Project of Southwestern University of Finance and Economics, and "Double-First Class" International Innovation Project (SYL22 GJCX08), and Fundamental Research Funds for the Central Universities (JBK2307022).

References

[1] R.C. Basole, M.A. Bellamy, Supply network structure, visibility, and risk diffusion: a computational approach, Decis. Sci. 45 (2014) 753–789, Wiley Online Library.
[2] N. Jegadeesh, S. Titman, Returns to buying winners and selling losers: implications for stock market efficiency, J. Finance 48 (1993) 65–91, Wiley Online Library.
[3] D. Wang, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, J. Zhou, S. Yang, Y. Qi, A semi-supervised graph attentive network for financial fraud detection, in: Proceedings of ICDM, 2019, pp. 598–607.
[4] B. Hu, Z. Zhang, C. Shi, J. Zhou, X. Li, Y. Qi, Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism, in: Proceedings of AAAI, 2019, pp. 946–953.
[5] S. Yang, Z. Zhang, J. Zhou, Y. Wang, W. Sun, X. Zhong, Y. Fang, Q. Yu, Y. Qi, Financial risk analysis for smes with graph-based supply chain mining, in: Proceedings of IJCAI, 2021, pp. 4661–4667.
[6] B. Hu, Z. Zhang, J. Zhou, J. Fang, Q. Jia, Y. Fang, Q. Yu, Y. Qi, Loan default analysis with multiplex graph learning, in: Proceedings of CIKM, 2020, pp. 2525–2532.
[7] D. Wang, Z. Zhang, J. Zhou, P. Cui, J. Fang, Q. Jia, Y. Fang, Y. Qi, Temporal-aware graph neural network for credit risk prediction, in: Proceedings of SDM, 2021, pp. 702–710.
[8] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, Q. He, Pick and choose: a GNN-based imbalanced learning approach for fraud detection, in: Proceedings of WWW'2021, 2021, pp. 3168–3177.
[9] C. Yin, C. Jiang, H.K. Jain, Z. Wang, Evaluating the credit risk of smes using legal judgments, Decis. Support Syst. 136 (2020) 113364, Elsevier.
[10] U. Ali, D. Hirshleifer, Shared analyst coverage: unifying momentum spillover effects, J. Financ. Econ. 136 (2020) 649–675, Elsevier.
[11] S. Lee, W.S. Choi, A multi-industry bankruptcy prediction model using back-propagation neural network and multivariate discriminant analysis, Expert Syst. Appl. 40 (2013) 2941–2946, Elsevier.
[12] A. Alonso Robisco, J.M. Carbo Martinez, Measuring the model risk-adjusted performance of machine learning algorithms in credit default prediction, Financ. Innov. 8 (2022) 1–35, Springer.
[13] D.L. Olson, D. Delen, Y. Meng, Comparative analysis of data mining methods for bankruptcy prediction, Decis. Support Syst. 52 (2012) 464–473, Elsevier.
[14] B.E. Erdogan, S. Özöugür-Akyüz, P.K. Atacs, A novel approach for panel data: an ensemble of weighted functional margin svm models, Inf. Sci. 557 (2021) 373–381, Elsevier.
[15] T. Hosaka, Bankruptcy prediction using imaged financial ratios and convolutional neural networks, Expert Syst. Appl. 117 (2019) 287–299, Elsevier.
[16] P.A. Borochin, J.E. Cicon, R.J. DeLisle, S.M. Price, The effects of conference call tones on market perceptions of value uncertainty, J. Financ. Mark. 40 (2018) 75–91, Elsevier.
[17] J. Li, L. Yang, B. Smyth, R. Dong, Maec: a multimodal aligned earnings conference call dataset for financial risk prediction, in: Proceedings of CIKM, 2020, pp. 3063–3070.
[18] C. Cai, L. Pan, X. Li, S. Luo, Z. Wu, A risk identification model for ICT supply chain based on network embedding and text encoding, Expert Syst. Appl. 228 (2023) 120459, Elsevier.
[19] H. Elsinger, A. Lehar, M. Summer, Risk assessment for banking systems, Manag. Sci. 52 (2006) 1301–1314, INFORMS.
[20] D. Acemoglu, A. Ozdaglar, A. Tahbaz-Salehi, Systemic risk and stability in financial networks, Am. Econ. Rev. 105 (2015) 564–608, American Economic Association.
[21] G. Kou, Y. Xu, Y. Peng, F. Shen, Y. Chen, K. Chang, S. Kou, Bankruptcy prediction for SMEs using transactional data and two-stage multiobjective feature selection, Decis. Support Syst. 140 (2021) 113429, Elsevier.
[22] X. Xie, X. Shi, J. Gu, X. Xu, Examining the contagion effect of credit risk in a supply chain under trade credit and bank loan offering, Omega 115 (2023) 102751, Elsevier.
[23] J. Long, C. Jiang, S. Dimitrov, Z. Wang, Clues from networks: quantifying relational risk for credit risk evaluation of SMEs, Financ. Innov. 8 (2022) 1–41, Springer.
[24] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of ICLR, 2017.
[25] Y. Yang, Z. Guan, J. Li, W. Zhao, J. Cui, Q. Wang, Interpretable and efficient heterogeneous graph convolutional network, IEEE Trans. Knowl. Data Eng. 35 (2023) 1637–1650, IEEE.
[26] Y. Xie, Y. Liang, M. Gong, A. Qin, Y.-S. Ong, T. He, Semisupervised graph neural networks for graph classification, IEEE Trans. Cybern. (2022), IEEE.
[27] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P.S. Yu, Heterogeneous graph attention network, in: Proceedings of WWW'2019, 2019, pp. 2022–2032.
[28] Y. Zhao, H. Zhou, A. Zhang, R. Xie, Q. Li, F. Zhuang, Connecting embeddings based on multiplex relational graph attention networks for knowledge graph entity typing, IEEE Trans. Knowl. Data Eng. 35 (2022) 4608–4620, IEEE.
[29] J. Yang, J. Lu, S. Lee, D. Batra, D. Parikh, Graph R-CNN for scene graph generation, in: Proceedings of ECCV, 2018, pp. 670–685.
[30] Y. Liu, M. Jin, S. Pan, C. Zhou, Y. Zheng, F. Xia, S.Y. Philip, Graph self-supervised learning: a survey, IEEE Trans. Knowl. Data Eng. 35 (2022) 5879–5900, IEEE.
[31] D. Cheng, Z. Niu, Y. Zhang, Contagious chain risk rating for networked-guarantee loans, in: Proceedings of SIGKDD, 2020, pp. 2715–2723.
[32] Y. Zheng, V.C. Lee, Z. Wu, S. Pan, Heterogeneous graph attention network for small and medium-sized enterprises bankruptcy prediction, in: Proceedings of PAKDD, 2021, pp. 140–151.
[33] W. Bi, B. Xu, X. Sun, Z. Wang, H. Shen, X. Cheng, Company-as-tribe: company financial risk assessment on tribe-style graph with hierarchical graph neural networks, in: Proceedings of SIGKDD, 2022, pp. 2712–2720.
[34] D. Cheng, Z. Niu, J. Zhang, Y. Zhang, C. Jiang, Critical firms prediction for stemming contagion risk in networked-loans through graph-based deep reinforcement learning, in: Proceedings of AAAI, 2023, pp. 14205–14213.
[35] X. Sun, H. Yin, B. Liu, H. Chen, J. Cao, Y. Shao, N.Q. Viet Hung, Heterogeneous hypergraph embedding for graph classification, in: Proceedings of WSDM, 2021, pp. 725–733.

[36] Y. Zeng, Q. Jin, T. Bao, W. Li, Multi-modal knowledge hypergraph for diverse image retrieval, in: Proceedings of AAAI, 2023, pp. 3376–3383.
[37] Y. Ma, Z. Guo, Z. Ren, J. Tang, D. Yin, Streaming graph neural networks, in: Proceedings of SIGIR, 2020, pp. 719–728.
[38] Y. Feng, H. You, Z. Zhang, R. Ji, Y. Gao, Hypergraph neural networks, in: Proceedings of AAAI, 2019, pp. 3558–3565.
[39] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of ICML, 2015, pp. 448–456.
[40] D. Hendrycks, K. Gimpel, Gaussian error linear units (GELUS), arXiv preprint, arXiv:1606.08415, 2016.
[41] W. Shang, K. Sohn, D. Almeida, H. Lee, Understanding and improving convolutional neural networks via concatenated rectified linear units, in: Proceedings of ICML, 2016, pp. 2217–2225.
[42] D.W. Hosmer Jr, S. Lemeshow, R.X. Sturdivant, Applied Logistic Regression, Wiley Series in Probability and Statistics, vol. 398, 2013.
[43] J.A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (1999) 293–300, Springer.
[44] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232, JSTOR.
[45] P. Velivcković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: Proceedings of ICLR, 2018.
[46] M. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: Proceedings of ESWC, 2018, pp. 593–607.
[47] C. Zhang, D. Song, C. Huang, A. Swami, N.V. Chawla, Heterogeneous graph neural network, in: Proceedings of SIGKDD, 2019, pp. 793–803.
[48] Y. Dong, N.V. Chawla, A. Swami, metapath2vec: scalable representation learning for heterogeneous networks, in: Proceedings of SIGKDD, 2017, pp. 135–144.
[49] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, C. Zou, Hypergraph learning: methods and practices, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2020) 2548–2566, IEEE.
[50] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint, arXiv:1412.6980, 2014.
[51] I. Loshchilov, F. Hutter, SGDR: stochastic gradient descent with warm restarts, in: ICLR, 2017, pp. 1–16.
[52] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605, JMLR.