Neural Networks 169 (2024) 475–484

Contents lists available at ScienceDirect

Neural Networks

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

Full Length Article

Enhancing Enterprise Credit Risk Assessment with Cascaded Multi-level Graph Representation Learning

Lingyun Song *, Haodong Li, Yacong Tan, Zhanhuai Li, Xuequn Shang

School of Computer Science, Northwestern Polytechnical University, Xi'an, 710129, China

Key Laboratory of Big Data Storage and Management, Northwestern Polytechnical University, Ministry of Industry and Information Technology, Xi'an, 710129, China

A R T I C L E I N F O

Keywords:
Knowledge graph
HyperGraph neural networks
Deep learning

A B S T R A C T

The assessment of Enterprise Credit Risk (ECR) is a critical technique for investment decisions and financial regulation. Previous methods usually construct enterprise representations by credit-related indicators, such as liquidity and staff quality. However, indicators of many enterprises are not accessible, especially for the small- and medium-sized enterprises. To alleviate the indicator deficiency, graph learning based methods are proposed to enhance enterprise representation learning by the neighbor structure of enterprise graphs. However, existing methods usually only focus on pairwise relationships, and overlook the ubiquitous high-order relationships among enterprises, e.g., supply chain connecting multiple enterprises. To resolve this issue, we propose a Multi-Structure Cascaded Graph Neural Network framework (MS-CGNN) for ECR assessment. It enhances enterprise representation learning based on enterprise graph structures of different granularity, including knowledge graphs of pairwise relationships, homogeneous and heterogeneous hypergraphs of high-order relationships. To distinguish influences of different types of hyperedges, MS-CGNN redefine new type-dependent hyperedge weight matrices for heterogeneous hypergraph convolutions. Experimental results show that MS-CGNN achieves state-of-the-art performance on real-world ECR datasets.

1. Introduction

Being an important technique for maintaining financial stability, Enterprise Credit Risk (ECR) assessment has received extensive attention from financial and artificial intelligence communities (Tang et al., 2021). The credit ratings of enterprises published by professional rating agencies are widely adopted by many financial investment and regulatory agencies. However, the credit assessment of traditional rating agencies is a time-consuming work, which relies on experienced experts and a large number of financial and non-financial indicator data. This limits these agencies to focus only on famous large companies, overlooking ubiquitous Small- and Medium-sized Enterprises (SMEs).

In recent years, some researchers try to improve the efficiency of the ECR assessment by statistical machine learning method, e.g., random forests (Fang, Wu, Zhu, & Xie, 2010; Li & Zhu, 2019). These works usually model the assessment of ECR as a classification task and construct enterprise representations by financial and non-financial indicators, such as asset–liability ratio or staff quality. However, many SME do not publish detailed financial or non-financial indicators, which makes it difficult for these methods to construct eligible enterprise representations. Besides, these methods overlook the interactions among enterprises (e.g., investment or supply relationships) that contain valuable information for ECR assessment (Yang et al., 2021).

To address this issue, some recent graph learning based ECR assessment methods enhance enterprise representation learning by the interactions from neighbor structure of enterprise graphs. However, these methods only focus on pairwise enterprise relationships of a single type (Cheng et al., 2019; Yang et al., 2021). This impedes the performance improvement due to the following two reasons:

(1) Enterprise interactions are much more complex than pairwise in many financial scenarios. For example, supply-chain graphs have been proved to be important for ECR assessment (Yang et al., 2021) and one of its toy example is shown in Fig. 1.

* Corresponding author at: School of Computer Science, Northwestern Polytechnical University, Xi'an, 710129, China.
E-mail addresses: lysong@nwpu.edu.cn (L. Song), lihaodong@mail.nwpu.edu.cn (H. Li), tanyacong@mail.nwpu.edu.cn (Y. Tan), lizhh@mail.nwpu.edu.cn (Z. Li), shang@nwpu.edu.cn (X. Shang).

https://doi.org/10.1016/j.neunet.2023.10.050
Received 9 June 2023; Received in revised form 24 September 2023; Accepted 30 October 2023
Available online 3 November 2023
0893-6080/

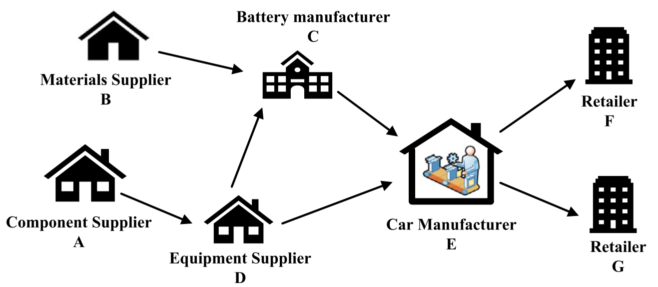

Fig. 1. A toy example of the supply chain connecting multiple enterprises in the electric vehicle industry.

It is ill-advised to model the supply interactions among enterprises using simple edges connecting pairwise objects due to two reasons: (i) The electric vehicle production and sales process expressed by the supply chain simultaneously involves multiple enterprises, leading to high-order enterprise relationships. (ii) Once an enterprise encounters credit risks, the risks can spread to other enterprises in the supply-chain graph, regardless of whether these enterprises are directly connected to the enterprise in the risk (Zhao, Chen, Wang, & Han, 2018).

(2) The interactions among enterprises are usually characterized by multiple levels, such as pairwise and high-order. However, existing methods usually only consider one type of interactions, which makes it difficult for them to enrich enterprise representations effectively by topology structure information at different levels of granularity.

To address the above issues, we propose to assess the ECR by a Multi-Structure Cascaded Graph Neural Network (MS-CGNN). It enhances the learning of enterprise representations by both pairwise and high-order topological structures of enterprise graphs, which are derived from the equity and supply relationships among enterprises. The main contributions of our work are presented as follows:

- We propose a new cascaded graph neural network framework for ECR assessment, which learns multi-level graph convolutional features capturing rich pairwise and high-order topology structure information. These graph structures at different levels of granularity characterize multiple financial interactions among enterprises and non-financial indicators of enterprises. That is, these graph structures of different levels carry semantic information at different granularity that is complementary to each other, which helps to improve the learning of enterprise representations.
- We propose a new hyperedge-type dependent attention mechanism for heterogeneous hypergraph convolutions, which assigns different weights to each type of enterprise hyperedge adaptively. This enables MS-CGNN to explicitly distinguish the influences of different type of hyperedges, emphasizing the information from the types of hyperedges important to the ECR assessment task.
- We conduct extensive experiments on real-world ECR assessment datasets. Experimental results show that MS-CGNN can achieve state-of-the-art performance.

2. Related work

2.1. ECR assessment

ECR assessment methods can be roughly divided into two groups: *Statistical learning based*, *Deep learning based* and *Graph learning based*.

Statistical learning based methods: This group of methods construct probability statistical models based on observable enterprises' characteristics to assess the ECR. Early ECR methods are mainly based on Linear Discriminant Analysis (LDA) and logistic regression. For example, the LDA based assessment methods (Altman, Iwanicz-Drozdowska, Laitinen, & Suvas, 2017) use a linear discriminant function to classify enterprises into two categories based on enterprises' credit-related characteristics. Relaxing some of the assumptions for LDA, logistic regression based methods estimate continuous ECR probabilities (Wang, 2021; Zhang & Zhang, 2016). Although achieving promising performance, the aforementioned methods have limitations in modeling nonlinear interactions between enterprise characteristics and credit risks.

To address this issue, some other statistic machine learning methods are developed, such as SVM (Hu, Hu, Chen, & Li, 2021; Zhang, Hu, & Zhang, 2015) based and Random Forest (Arora & Kaur, 2020) based methods. For example, Zhang et al. (2015) proposed to assess the ECR in supply chain finance via a support vector machine model that is constructed by 28 indicators of four aspects, including industry status, SME credit conditions, leading enterprise credit conditions and the status of cooperative relationships. Arora and Kaur (2020) construct random forest based classifiers for credit assessment based on the enterprise features selected by bootstrap-lasso. Besides, some other researchers develop ensemble methods to make a comprehensive ECR estimation. For example, Yao, Hu, and Wang (2022) proposed a SVM ensemble model for ECR assessment in the supply chain context. Zhu, Zhou, Xie, et al. (2019) proposed an enhanced hybrid ensemble ECR estimation method that incorporates two classic ensemble approaches, i.e., random subspace and multiboosting. However, these methods have strong dependence on enterprises' financial and non-financial characteristics, such as *Current ratio* and *Asset–liability ratio*, which are difficult to acquire for many enterprises, especially SMEs. This limits the application of these statistical learning based methods.

Deep learning based methods: Due to the powerful abilities in representation learning and interaction modeling, deep neural networks (Bacciu, Errica, Micheli, & Podda, 2020; Schmidhuber, 2015) are used as classifiers to predict the probability of ECR. For example, Derelioğlu, Gürgen, and Okay (2009) proposed to assess the ECR by a cascaded MLP classifier (Lengellé & Denoeux, 1996) based on a set of selected enterprise characteristics. Later, researchers proposed to improve the accuracy of ECR classification by modeling the internal correlations between different dimensions of enterprise features (Zhang, Wang, Chen, Shang, & Tian, 2017). The applications of above methods still are limited since the training of neural classifiers also require large amounts of enterprise characteristics that are hard to collect. Besides, the aforementioned methods overlook the relations between enterprises, which reduces their abilities to accurately assess credit risk (Wang, Zhang, et al., 2021). This is because an enterprise may fall into credit risk due to the abnormalities of its related enterprises (e.g., supplier enterprises), even if its credit-related indicators are normal. Thus, the credit-related characteristics of an enterprise also have an impact on the credit status of its related enterprises.

Graph learning based methods: To deal with sparse enterprise characteristics mentioned above, researchers proposed to enhance ECR assessment by graph learning methods, which enrich the representation of an enterprise by integrating the characteristic information of its neighbors in enterprise graphs. In recent years, mainstream graph learning methods are based on various graph neural networks (Jiang, Chen, Wang, Xu, & Luo, 2023; Li, Ma, Wang, & Zhuang, 2020; Phan, Nguyen, Nguyen, & Bui, 2018; Veličković et al., 2017). For example, the problem of data deficiency in SME representation learning is tackled by the topology structure of supply-chain graphs (Yang et al., 2021). To capture rich semantics of enterprises in scenarios, graph structures of different views generated by multi-view graph learning are encoded into enterprise representations (Wang, Yu, Zhang, & Zhang, 2021). Although these methods achieve promising performance, they only take into account pairwise topology structure of enterprise graphs, overlooking the high-order relationships among enterprises.

Besides, existing methods ignore the fact that the semantics conveyed by topology structures of different level are complementary. This makes it difficult for these methods to comprehensively capture the semantics of enterprise graphs and thus reduces the semantic representation ability of the learned enterprise features. By contrast, our proposed MS-CGNN improves enterprise representation learning for ECR assessment by simultaneously combining enterprise graph structures of different granularity via a deep cascaded neural framework, including both enterprise graphs of pairwise and high-order relationships.

2.2. Hypergraph learning

In real word, the interactions among entities are often not pairwise but rather triadic or even higher (Liu, Song, Wang, & Shang, 2023; Sun et al., 2021), such as “multiple authors co-authoring a paper” and “multiple papers published in a venue", which are difficult to be modeled by a simple edge connecting two vertices. To address this issue, in recent years many efforts have been devoted to modeling these high-order correlations among data by hypergraph neural networks. They encode high-order topology structures of hyperedges connecting multiple vertices into vertex representations by hyperedge convolution operations (Feng, You, Zhang, Ji, & Gao, 2019). Later, hypergraph attention mechanisms (Bai, Zhang, & Torr, 2021; Chen, Cheng, Li, & Wang, 2020) are proposed for hypergraph convolutions, where different hyperedge and vertices are assigned different aggregation weights in attentive feature aggregations. The aforementioned methods focus on single hypergraph learning and usually use a pre-concatenation approach when confronting multiple hypergraphs. This ignores the inter-correlations among hypergraphs and the differences between different modality vertices.

To address this issue, heterogeneous hypergraph learning (Huang, Huang, & Yang, 2021; Sun et al., 2021; hu, Zhao, Hu, & Gao, 2019) has been studied for the hypergraphs containing multiple types of vertices or relationships. For example, some researchers propose multi-hypergraph learning methods (Huang et al., 2021; hu, Zhao, Hu, & Gao, 2019), where multiple distinct branches of hypergraph learning are performed in parallel to separately extract high-level information from each modality. By the homogeneous hypergraph snapshots extracted from heterogeneous hypergraphs, heterogeneous hypergraph learning is decomposed into multiple homogeneous hypergraph learning (Sun et al., 2021). The information learned from these different hypergraph snapshots are aggregated to generate comprehensive representations. Although achieving promising performance, existing methods do not explicitly distinguish the influence of hyperedges of different types in the hyperedge aggregation. This may result in illogical hyperedge weights that impede the performance of representation learning. By contrast, our MS-CGNN use a new hyperedge-type dependent attention mechanism, which can adaptively emphasize the information from the types of hyperedges important to the ECR assessment. Besides, existing hypergraph learning methods only focus on learning vertex embedding based on high-order graph structures, ignoring the semantics of higher-order and pairwise graph structures are complementary.

3. Methodology

Before diving into the proposed MS-CGNN, we introduce some notations and definitions first. Then, we briefly introduce enterprise knowledge graph and hypergraph construction. Finally, we present the technical details about our proposed MS-CGNN.

3.1. Notations and definitions

Definition 1 (*Enterprise Knowledge Graph*)**.** An enterprise knowledge graph is a directed graph composed of enterprise entities as vertexs and enterprise relationships as different types of edges. An instance of edge is a fact triplet denoted by (h, r, t), where $h \in E$, $r \in R$ and $t \in E$ denote the head, relation, and tail of the fact triplet. E is the set of enterprise entities and R is a set of pairwise relationships between enterprise entities.

Definition 2 (*Enterprise Hypergraph*)**.** An enterprise hypergraph can be defined as $G = \{V, E, W\}$, where V is a set of enterprise vertices and E is a set of enterprise hyperedges representing the high-order relationships among two or more enterprises. $W \in \mathbb{R}^{|E|\times|E|}$ is a diagonal weight matrix of hyperedges. For any hyperedge $e \in E$, it can be denoted as $\{v_i, v_j, \dots, v_k\} \subseteq V$. In this paper, we use $H = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{|V|}\}$ to denote the vertex embedding matrix and $F = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{|E|}\}$ to denote the hyperedge embedding matrix. The topological structure of a hypergraph G can also be represented by an incidence matrix $H \in \mathbb{R}^{|V|\times|E|}$, with entries defined as:

$$h(v, e) = \begin{cases} 1, & \text{if } v \in e. \\ 0, & \text{if } v \notin e. \end{cases} \tag{1}$$

Definition 3 (*Heterogeneous Hypergraph*)**.** A heterogeneous hypergraph can be defined as $G = \{V, E, T_v, T_e, W\}$. Here V is a set of vertices and T_v is a set of vertex types. E is a set of hyperedges and T_e is a set of hyperedge types. When $|T_v| + |T_e| > 2$, the hypergraph is heterogeneous.

3.2. Enterprise knowledge graph construction

In this paper, we construct two enterprise knowledge graphs by two types of interactions between enterprises, i.e., equity relationships and supply chain relationships. For example, for the equity knowledge graph, we construct a fact triplet by taking a shareholder enterprise as the head entity, the shareholding as the relationship, and a invested enterprise as the tail entity. Similarly, for the supply-chain knowledge graph, we construct the fact triplets by taking enterprises as the head/tail entities, and the supplier/customer relationships as the relationships between the head and end entities. In this paper, we denote the equity knowledge graph by KG_Q, and denote supply-chain knowledge graph by KG_S.

3.3. Enterprise hypergraph construction

As shown in Fig. 2, we first construct two homogeneous enterprise hypergraphs according to equality and supply-chain relationships among enterprises, respectively. Then, we construct a heterogeneous enterprise hypergraph using non-financial attributes of enterprises.

3.3.1. Equity hypergraph construction

For each enterprise vertex v_i, we generate an equity hyperedge that is composed of v_i and all the enterprises investing in v_i. The generated equity hypergraph is denoted as $G_Q = \{V_Q, E_Q, W_Q\}$, where V_Q is a vertex set and E_Q is a hyperedge set. $W_Q \in \mathbb{R}^{|E_Q|\times|E_Q|}$ is a weight matrix of equity hyperedges and usually is initialized to an identity matrix I.

3.3.2. Supply-chain hypergraph construction

For each enterprise vertex v_i, we generate a supply-chain hyperedge that is composed of v_i and all the enterprises involved in the transaction of the supply chain of v_i. The generated supply-chain hypergraph is denoted as $G_S = \{V_S, E_S, W_S\}$. The hyperedge weight matrix $W_S \in \mathbb{R}^{|E_S|\times|E_S|}$ is initialized to an identity matrix I.

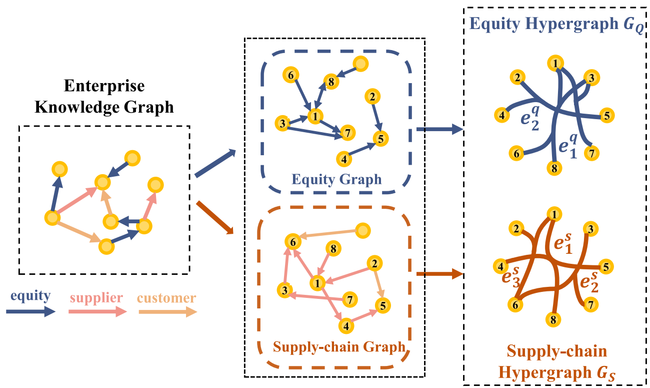

Fig. 2. Hypergraph construction based on the equity and supply-chain relationships between enterprises.

3.3.3. Attribute hypergraph construction

Enterprises vertices also contain rich attributes that have been demonstrated helpful for ECR assessment, e.g., performance status and industry. Thus, in this part, we also construct a heterogeneous attribute hypergraph that contain two types of hyperedges. Specifically, one type hyperedge connects all the enterprises in the same industry, and the other type of hyperedge connects all the enterprise with the same performance status. The generated heterogeneous attribute hypergraph is denoted as $G_A = \{V_A, E_A, T_v^A, T_e^A, W_A\}$, where $W_A \in \mathbb{R}^{|E_A|\times|E_A|}$ is a diagonal weight matrix of hyperedges and is initialized to an identity matrix I.

3.4. Multi-structure cascaded graph neural network (MS-CGNN)

In Fig. 3, we show the architecture of the proposed MS-CGNN, which is a cascade GCN framework that incorporate multiple graph structures at different level of granularity into graph convolution based representation learning. As shown in the figure, MS-CGNN are comprised of three component network: (i) Knowledge Graph Embedding Network (KGEN). (ii) Multi-Hypergraph Attention Network (MHAN). (iii) Hyperedge-type Dependent Attention Network (HDAN).

Given the enterprise graphs characterizing pair-wise equity and supply interactions, MS-CGNN first learns base representations for each enterprise by the KGEN, which integrates credit-related financial features from its neighboring enterprises of the graphs. Then, MS-CGNN further learn to enrich the base enterprise representations by a MHAN, which uses two distinct homogeneous hypergraph convolution networks to separately encode homogeneous high-order enterprise relationship information of equity and supply-chain hypergraphs, respectively. Finally, the enterprise representations learned from above two hypergraph convolutions are combined and fed into the HTAN, which aims to encode the heterogeneous high-order relationships derived from non-financial enterprise attributes into enterprise representations.

3.4.1. Enterprise knowledge graph embedding network

In this part, we first learn base enterprise representations based on two fundamental pairwise directed relationships between enterprises, i.e., equity and supply relationships. Specifically, as shown in Fig. 3, we use the knowledge graph embedding network (Wang, Zhang, Feng, & Chen, 2014) to learn relation-specific enterprise representations given the equity graph KG_Q and the supply graph KG_S. The enterprise representation matrices learned from the equity and supply graphs are denoted by $X^q = \{\mathbf{x}_1^q, \mathbf{x}_2^q, \dots, \mathbf{x}_{|V|}^q\} \in \mathbb{R}^{|V|\times D_k}$ and $X^s = \{\mathbf{x}_1^s, \mathbf{x}_2^s, \dots, \mathbf{x}_{|V|}^s\} \in \mathbb{R}^{|V|\times D_k}$, respectively. The $|V|$ denotes the number of enterprise vertices and D_k is the size of the enterprise representations learned by KGEN.

3.4.2. Multi-hypergraph attention network

In this part, we further use a Multi-Hypergraph Attention Network (MHAN) to enrich the learned base enterprise representations by the interactions beyond pairwise relations among enterprises. First, MHAN employs two branches to learn enterprise representations capturing the high-order topology structures of the equity hypergraph G_Q and the supply-chain hypergraph G_S, respectively. Second, MHAN combines the enterprise representations learned from above two hypergraphs to learn comprehensive representations capturing rich semantics.

Specifically, in each branch, one hypergraph (G_Q or G_S) is fed into a Hypergraph Attention Network (HAN) to perform homogeneous hypergraph convolutions. As shown in Fig. 4, each layer of HAN learns enterprise representations by two modules: attentive vertex aggregation and attentive hyperedge aggregation.

(1) Attentive Vertex Aggregation. This module aims to learn the representations of hyperedges $e_j \in E$, denoted by $\mathbf{f}_j^l$. As different enterprise vertices contribute differently to equity or supply-chain hyperedges e_j, the lth layer of HAN learns the representations for e_j by Eq. (2), which aggregates the connected enterprise vertex information by a vertex-level attention.

$$\mathbf{f}_j^l = \sigma(\sum_{v_k\in e_j} c_{jk}^v \mathbf{h}_k^{l-1} P), \tag{2}$$

where $\mathbf{h}_k^0 \in \{\mathbf{x}_k^q, \mathbf{x}_k^s\}$ and σ is a nonlinearity function such as $LeakyReLu$. $P \in \mathbb{R}^{D_{(l-1)}\times D_l}$ is a learnable weight matrix, where $D_{(l-1)}$ is the output size of the $(l-1)$-th layer and D_l is the output size of the lth layer. The c_{jk}^v denotes the attention coefficient of vertex v_k on the hyperedge e_j, quantifying how important the v_k is to the e_j. Specifically, c_{jk}^v can be computed by

$$c_{jk}^v = \frac{exp\left(\sigma([\mathbf{h}_k^{l-1}P \parallel \mathbf{f}_j^{l-1}P]\mathbf{a}_v)\right)}{\sum_{v_t\in e_j} exp\left(\sigma([\mathbf{h}_t^{l-1}P \parallel \mathbf{f}_j^{l-1}P]\mathbf{a}_v)\right)}, \tag{3}$$

where $\mathbf{h}_k^{l-1} \in \mathbb{R}^{1\times D_{(l-1)}}$ and $\mathbf{f}_j^{l-1} \in \mathbb{R}^{1\times D_{(l-1)}}$ denote the vertex embedding and hyperedge feature from the $(l-1)$-th layer, respectively. The $\mathbf{a}_v \in \mathbb{R}^{2*D_l\times 1}$ is a learnable weight vector used to output a scalar similarity value. The symbol $\parallel$ denotes the concatenation operation. σ is the nonlinearity such as $LeakyReLu$,

(2) Attentive Hyperedge Aggregation. We denote the set of hyperedges connecting to vertex v_i as $\varepsilon_i = \{e_1, \dots, e_m\}$. Given all the hyperedge representations $\{\mathbf{f}_j^l|\forall e_j \in \varepsilon_i\}$, this module learns the representations of v_i (i.e., $\mathbf{h}_i$) by Eq. (4), which aggregates all the information of its connected hyperedges via a hyperedge-level attention.

$$\mathbf{h}_i^l = \sigma(\sum_{e_j\in\varepsilon_i} c_{ij}^e \mathbf{f}_j^l), \tag{4}$$

where c_{ij}^e denotes the attention coefficient of hyperedge e_j on vertex v_i, highlighting the informative hyperedges for learning the next-layer representation of vertex v_i. The c_{ij}^e can be computed by

$$c_{ij}^e = \frac{exp\left(\sigma([\mathbf{f}_j^l \parallel \mathbf{h}_i^{l-1}P]\mathbf{a}_e)\right)}{\sum_{e_t\in\varepsilon_i} exp\left(\sigma([\mathbf{f}_t^l \parallel \mathbf{h}_i^{l-1}P]\mathbf{a}_e)\right)}, \tag{5}$$

where σ is the nonlinearity such as $LeakyReLu$, and $\mathbf{a}_e \in \mathbb{R}^{2*D_l}$ is a shared weight vector. The enterprise representation matrix learned from the equity hypergraph and supply-chain hypergraph by $H^q = \{\mathbf{h}_1^q, \mathbf{h}_2^q, \dots, \mathbf{h}_{|V|}^q\} \in \mathbb{R}^{|V|\times D_l}$ and $H^s = \{\mathbf{h}_1^s, \mathbf{h}_2^s, \dots, \mathbf{h}_{|V|}^s\} \in \mathbb{R}^{|V|\times D_l}$, respectively. $|V|$ denotes the number of enterprise vertices, D_l denotes the dimensionality of the enterprise vertex representations $\mathbf{h}$ output by a MHAN with l layers.

Attentive Combination of H^q and H^s: In this part, we combine the H^q and H^s to learn comprehensive enterprise representations $H^c = \{\mathbf{h}_1^c, \mathbf{h}_2^c, \dots, \mathbf{h}_{|V|}^c\} \in \mathbb{R}^{|V|\times D_l}$, where $\mathbf{h}_i^c$ denotes the representation of the ith enterprise. Specifically, we learn $\mathbf{h}_i^c$ by integrating the enterprise

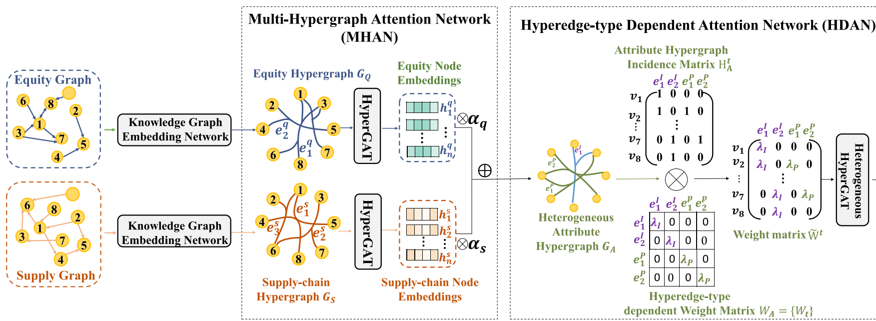

Fig. 3. Framework of the Enterprise Credit Risk Assessment based on Multi-Hypergraph Attention Network.

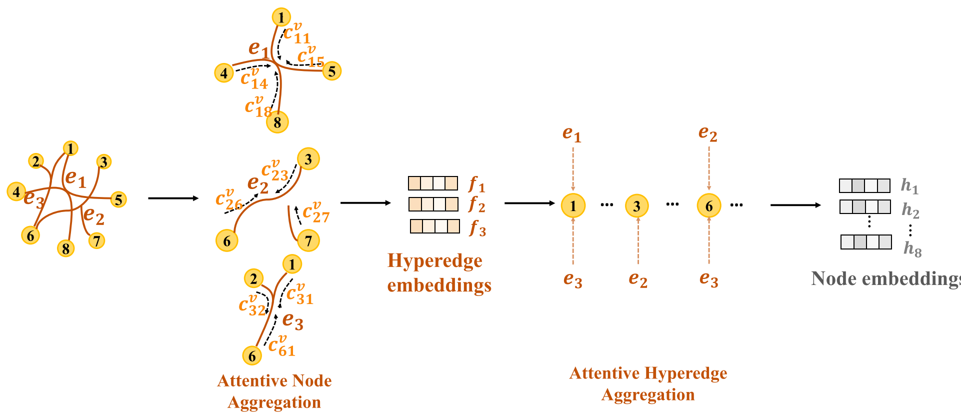

Fig. 4. Illustration of each layer of the hypergraph attention network, which learns representations of enterprise vertices by attentive vertex aggregation and attentive hyperedge aggregation.

representations learned from hypergraphs G_Q and G_S, which can be formulated by

$$\mathbf{h}_i^c = \alpha_i^q \cdot \mathbf{h}_i^q + \alpha_i^s \cdot \mathbf{h}_i^s, 1 \le i \le |V|, \tag{6}$$

where α_i^q and α_i^s denote the attention coefficients of $\mathbf{h}_i^q$ and $\mathbf{h}_i^s$, respectively. Specifically, the above attention coefficients can be learned by

$$\boldsymbol{\alpha}^q = \frac{exp\left(\sigma\left(H^q W\right)\mathbf{a}_c\right)}{exp\left(\sigma\left(H^q W\right)\mathbf{a}_c\right) + exp\left(\sigma\left(H^s W\right)\mathbf{a}_c\right)}, \tag{7}$$

$$\boldsymbol{\alpha}^s = \frac{exp\left(\sigma\left(H^s W\right)\mathbf{a}_c\right)}{exp\left(\sigma\left(H^q W\right)\mathbf{a}_c\right) + exp\left(\sigma\left(H^s W\right)\mathbf{a}_c\right)}, \tag{8}$$

where $\boldsymbol{\alpha}^q = (\alpha_1^q, \alpha_2^q, \dots, \alpha_{|V|}^q)$ and $\boldsymbol{\alpha}^s = (\alpha_1^s, \alpha_2^s, \dots, \alpha_{|V|}^s)$. The σ denotes the LeakyReLu function, $W \in \mathbb{R}^{D_l \times D_a}$ denotes the transformation weight matrix and $\mathbf{a}_c \in \mathbb{R}^{D_a \times 1}$ denotes a learnable parameter vector.

3.4.3. Hyperedge-type dependent attention network

In this part, we enhance the learning of enterprise representations by incorporating the high-order topology structures of a heterogeneous enterprise attribute hypergraph. Different types of hyperedge in the attribute hypergraph usually have different influences on the vertex's semantics. However, previous hypergraph learning methods usually take one identity matrix as the weight matrix of hyperedges, treating the hyperedges of different types equally. Although these methods learn attention weights for each hyperedge, they do not explicitly distinguish the effects of different types of hyperedge in vertex representation learning. This may result in ill-advised weights for hypergraph convolutions. For example, hyperedges of the same type are assigned with significantly different weights, e.g., 0.8 vs 0.1.

To address this issue, we redefine the hyperedge weight matrix W_A for the attribute hypergraph $G_A = \{V_A, E_A, T_v^A, T_e^A, W_A\}$ by a new diagonal weight matrix $\mathcal{W}_A$, distinguishing the weights of each hyperedge type by different diagonal submatrices $\mathcal{W}_t \in \mathbb{R}^{m_t \times m_t} (1 \le t \le |T_e^A|)$. The $|T_e^A|$ denotes the number of hyperedge types, m_t denotes the number of hyperedges belonging to the tth type. The diagonal vector of $\mathcal{W}_t$ is denoted as $\lambda_t \in \mathbb{R}^{m_t}$ and all its entries are equal, i.e., λ_t, denoting the weight of each hyperedge of the tth type. In this way, hyperedges of different types are initialized with different

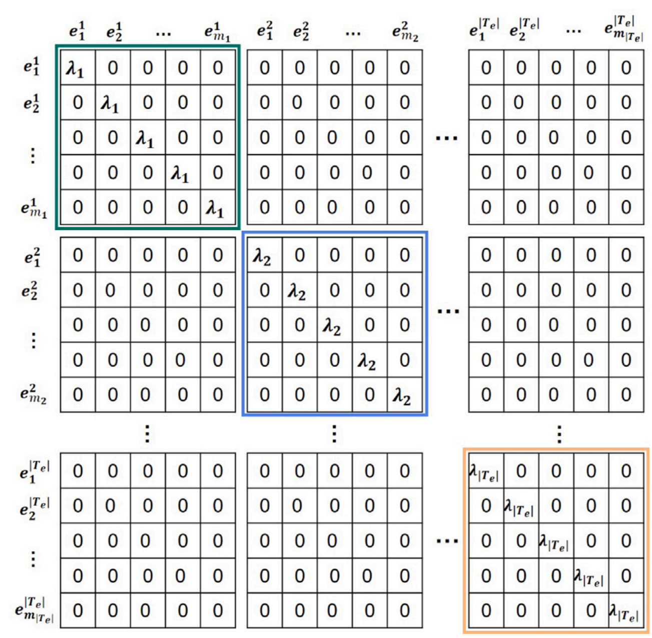

Fig. 5. The hyperedge weight matrix $\mathcal{W}_A$ for the heterogeneous attribute hypergraph.

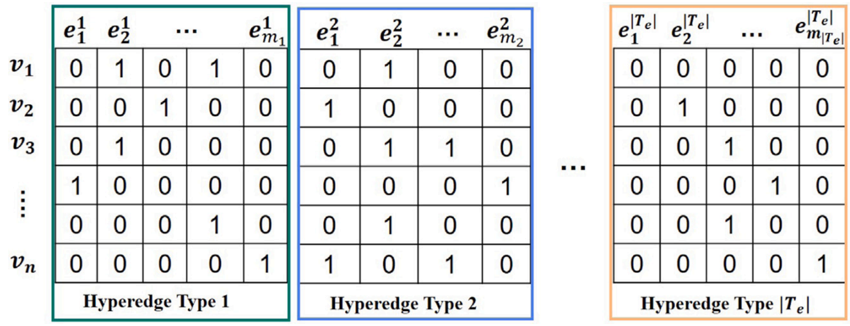

Fig. 6. The incidence matrix of heterogeneous attribute hypergraph, where each submatrix denotes the incidence matrix (i.e., H_A^t) for the tth type of hyperedge.

weights, while the hyperedges of the same type are initialized with the same weight. All the $\mathcal{W}_t$ are used to form the new hyperedge weight matrix $\mathcal{W}_A \in \mathbb{R}^{M\times M}$, where $M = \sum_{t=1}^{|T_e^A|} m_t$ and its diagonal vector is $(\lambda_1, \lambda_2, \dots, \lambda_{|T_e^A|}) \in \mathbb{R}^M$. Specifically, $\mathcal{W}_A$ can be illustrated by Fig. 5.

With the $\mathcal{W}_A$, we design a Hyperedge-type Dependent Attention Network (HDAN) to learn enterprise representations. In HDAN, Heterogeneous Hypergraph convolutions (Heterogeneous HyperConv) is performed by two modules, i.e., *attentive vertex aggregation* and *type-dependent hyperedge aggregation*.

(1) Attentive vertex aggregation. This module aims to learn the representations of an attribute hyperedge e_j by Eq. (2), which aggregates the information of the vertices of hyperedge e_j. The enterprise vertex representations of this module is initialized with the representations learned from Eq. (6) of the MHAN, i.e., H^c.

(2) Type-dependent hyperedge aggregation. In this module, by the new type-dependent hyperedge weight matrix $\mathcal{W}_A$, we learn vertex representations by a type-dependent hyperedge aggregation strategy, which explicitly distinguishes the influences of different types of hyperedges. Specifically, at the lth layer, we learn the representation $\mathbf{h}_j^l$ for hyperedge e_j by Eq. (9), which aggregates the information of its connected hyperedges by combining attention weights and hyperedge weights $\mathcal{W}_A$.

$$\mathbf{h}_j^l = \sigma\Big(\sum_{e_j\in\varepsilon_i} c_{ij}^e \tilde{w}_{ij}^t \mathbf{f}_j^l\Big) \tag{9}$$

where $\tilde{w}_{ij}^t$ denotes the weight of the jth hyperedge of the tth type on vertex v_i and all the weights $\tilde{w}_{ij}^t$ for the tth type form the matrix $\widetilde{W}^t = \{\tilde{w}_{ij}^t\}_{j=1}^{m_t} \in \mathbb{R}^{|V|\times m_t}$. The c_{ij}^e is the attention coefficients of hyperedge e_j on v_i and can be learned by Eq. (5). For the hyperedges of the same type, their weight on one vertex should be the same. Therefore, the $\widetilde{W}^t$ for the hyperedges of tth type on all vertices can be computed by

$$\widetilde{W}^t = \frac{H_A^t \mathcal{W}_t}{\left|E_i^t\right|}, \tag{10}$$

where $\mathcal{W}_t \in \mathbb{R}^{m_t\times m_t}$ denotes the redefined hyperedge weight matrix of the tth type. The $H_A^t \in \mathbb{R}^{|V|\times m_t}$ is a incidence matrix indicating the connections between the hyperedges of the tth type and all the vertices in G_A. $E_i^t = \{e_{ip}^t\}_{p=1}^{m_t^i}$ denotes the set of hyperedges of the tth type connected to v_i, m_t^i denotes the number of hyperedges in E_i^t, and e_{ip}^t denotes the pth hyperedge of the tth type connected to v_i. An example of H_A^t and $H_A^t\mathcal{W}_t$ are shown in Figs. 6 and 7, respectively. In Fig. 8, we illustrate the weight matrix $\widetilde{W}^t$ learned by Eq. (10). Through HDAN, we obtain the final representations of all enterprise vertices $H^a = \{\mathbf{h}_1^a, \mathbf{h}_2^a, \dots, \mathbf{h}_{|V|}^a\} \in \mathbb{R}^{|V|\times D}$, where $|V|$ is the number of enterprises and D is the dimensionality of the $\mathbf{h}_i^a$ learned by Module HDAN.

3.5. Enterprise credit risk assessment

In this part, we input the learned enterprise representation $\mathbf{h}_i^a$ into a prediction layer, which maps $\mathbf{h}_i^a$ to a continuous value between 0 and 1 that represents the probability score $\hat{y}_i$ for the ith enterprise having credit risk. Specifically, the computation of $\hat{y}_i$ can be formulated by

$$\hat{y}_i = \sigma\left(\mathbf{w}_i(\mathbf{h}_i^a)^\top + b_i\right), \tag{11}$$

where $\hat{y}_i$ represents the predicted risk probability for the ith enterprise. The $\mathbf{w}_i \in \mathbb{R}^{1\times D}$ denotes the learnable parameter vector of the prediction layer, $\mathbf{h}_i^a \in \mathbb{R}^{1\times D}$ denotes the representation of the ith enterprise learned by the final HDAN module, b_i is a bias and σ denotes the sigmoid function.

The cross-entropy loss is used to optimize the parameters of our proposed MS-CGNN. The cross-entropy loss is used to optimize the parameters of our proposed MS-CGNN, which can be formulated by

$$L = -\sum_{(x,y)\in D_{Train}} y log\left(\hat{y}\right) + (1-y)\, log\left(1-\hat{y}\right), \tag{12}$$

where D_{Train} is the training set, y denotes the ground truth and x represents the training samples.

Table 1
Statistics of the enterprises in our experimental datasets. n_1 denotes the number of all the enterprises in our dataset. n_2 denotes the number of enterprises with no indicators.

Type	n_1	n_2
A-share	4114	60
New Three Board	74	6
The Four Board	9	9
unlisted	998	998
Total	5197	1073

4. Experiments

4.1. Datasets

To the best of our knowledge, there is no available public datasets for ECR assessment. To verify the effectiveness of our MS-CGNN, we construct a new dataset by collecting real-world credit-related enterprise data. Specifically, we construct enterprise graphs by selecting 5195 companies in China, including 4114 companies listed in *A-share*, 74 companies listed in *New Three Board*, 9 companies listed in *The Four Board*, and 998 *unlisted companies*. The financial indicators of all the companies are not all available and the detailed statistics are shown in Table 1. Note that the data used in this research are processed by data abstraction and contains no identifiable information of companies.

Data Collection. We collect the credit-related financial indicators of the selected enterprises from the *Datayes*[1] and the *CSMAR*[2] databases. Specifically, as shown in Table 2, 18 financial indicators (Han, 2020) from four aspects, i.e., solvency, operations ability, profitability and

[1] http://uqer.datayes.com
[2] http://www.gtarsc.com

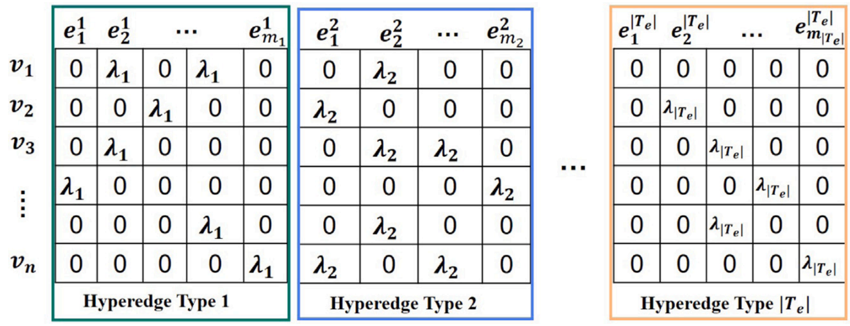

Fig. 7. The product results of the incidence matrix and the weight matrix, where each submatrix corresponds to one type of hyperedge.

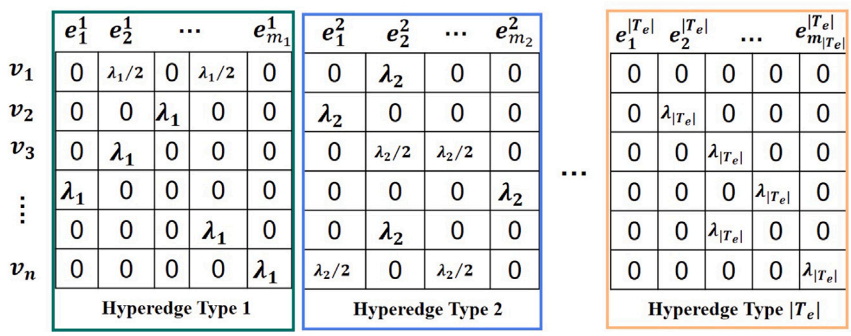

Fig. 8. An example of weight matrices $\widetilde{W}^t (1 \leq t \leq |T_e|)$ for $|T_e|$ types of hyperedges on vertices.

Table 2
The credit-related indicators for ECR assessment.

Indicator	Variable	Description	Definition
Solvency	X1	Current ratio	Current assets/current liabilities
	X2	Quick ratio	Quick assets/current liabilities
	X3	Equity ratio	Liabilities/owner's equity
	X4	Cash flow to debt ratio	Net cash flow from operating activities/liabilities
	X5	Asset liability rate	Total liabilities/total assets
	X6	Equity multiplier	Total assets/owner's equity
Operations ability	X7	Inventory turnover rate	Main business cost/average inventory balance
	X8	Accounts receivable turnover rate	Total operating revenue/average balance of accounts receivable
	X9	Current asset turnover rate	Total operating revenue/average total current assets
	X10	Fixed assets turnover rate	Main business revenue/average balance of fixed assets
	X11	Total assets turnover rate	Total operating revenue/average total assets
Profitability	X12	Return on equity	Net profit average balance of owner's equity
	X13	Net profit on total assets	Net profit/average balance of total assets
	X14	Net profit	Total profit minus income tax
Growth ability	X15	Operating revenue growth rate	Increase in operating revenue in the current period/operating income in the previous period
	X16	Net profit growth rate	Net profit increase in the current period/net profit in the previous period
	X17	Total assets growth rate	Total assets increase in the current period/total assets in the previous period
	X18	Net assets growth rate	Net assets increase in the current period/total net assets in the previous period

Table 3
Statistics of our experimental datasets. n_{edge} denotes the number of edges in each graph and n_{isov} denotes the number of isolated vertices in each graph.

Graph type	Edge type	n_{edge}	n_{isov}
Knowledge Graph	Equity triplets	4674	2468
	Supply triplets	17,478	1590
Homogeneous Hypergraphs	Equity Hyperedges	1782	2468
	Supply-chain Hyperedges	231,203	1590
Attribute Hypergraphs	Industry Hyperedges	875	0
	Performance status Hyperedges	8	0

Table 4
The performance results of all competing methods on the task of ECR Assessment.

	Model	Recall	F1-score	Accuracy
SML	Logistic Regression	0.0165	0.0324	0.7702
	XGBOOST	0.0370	0.0709	0.7731
GNN	GAT	0.5130	0.6374	0.8637
	GAT with KGEN	0.5226	0.6828	0.8865
	GraphSAGE	0.4000	0.5668	0.8469
	GraphSAGE with KGEN	0.4165	0.5788	0.8587
	RGCN	0.4435	0.5730	0.8209
	RGCN with KGEN	0.4733	0.6284	0.8692
HGNN	HGAT	0.5185	0.6829	0.8875
	HGAT with KGEN	0.5432	0.7021	0.8923
	MS-CGNN	**0.9300**	**0.8863**	**0.9442**

Table 5
The comparison results of the computation budget and the number of model parameters.

Model name	Running time	GPU Memory	Parameter number
MS-CGNN	6.78 s/Epoch	2015 MB	199,194
RGCN+KGEN	0.51 s/Epoch	1207 MB	187,670
GraphSAGE+KGEN	0.21 s/Epoch	1123 MB	210,410
GAT+KGEN	0.12 s/Epoch	1121 MB	187,786
HyperGAT+KGEN	1.30 s/Epoch	1839 MB	215,338

growth ability of enterprises, are used to constitute initial enterprise representations, i.e., 18-dim enterprise embeddings. The equity and supply relationships among enterprises are from the *CSMAR* database and the platform *Tianyan*.[3] The groundtruth of enterprise credit risk are collected from the *Tianyan*. In total, we have 1782 hyperedges in the equity hypergraph and 231,203 hyperedges in the supply chain hypergraph. The heterogeneous attribute hypergraph is constructed based on two types of enterprise attributes, i.e., enterprise industry and performance status, which are collected from the *CSMAR* database. Detailed statistics of the hyperedges in the above enterprise graphs and hypergraphs are shown in Table 3.

4.2. Experimental setup

Implementation Details. MS-CGNN is implemented by PyTorch and optimized with the Adam optimizer. The optimized learning rate α is set to 0.001 and the learning decay rate is set to 0.1. The learning rate α decays every 10 epochs using an exponential rate of 0.5. $L2$ regularization is applied to the parameters of our networks with $\lambda = 5e^{-4}$. A dropout with a rate of 0.3 is used for best performance. Before Module I, the representations of enterprises are initialized with 18 financial indicators or randomly initialized by a Gaussian distribution when the indicators are unavailable. The MHAN and HDAN modules of MS-CGNN both consist of two layers and the output of each module is a 18-dim representation vector. Specifically, we use the transformation matrices $P_1 \in \mathbb{R}^{18\times100}$ and $P_2 \in \mathbb{R}^{100\times100}$ for the first layer, and use $P_1 \in \mathbb{R}^{100\times18}$ and $P_2 \in \mathbb{R}^{18\times18}$ for the second layer. During the training phase, we train MS-CGNN for 100 epochs with early-stopping strategy. For baseline models, we use the codes provided by the authors using the parameters according to the best result in our dataset.

Baselines. We compare the performance of our MS-CGNN with previous state-of-art ECR assessment baselines, falling into three main groups: (i) Statistical Machine Learning (SML) Methods, i.e., popular Logistic regression (Xiong, Jia, Wenjie, et al., 2009) and XGBoost (Chen et al., 2015), which represents enterprises by credit-related indicators. (ii) GNN based methods, i.e., GAT (Veličković et al., 2017), GraphSAGE (Hamilton, Ying, & Leskovec, 2017) and RGCN (Schlichtkrull et al., 2018), which learns vertex representations based on pairwise relationships. (iii) HyperGraph Neural Network (HGNN) based methods, i.e., HGAT (Bai et al., 2021), which learns vertex representations based on all the high-order relationships, i.e., equity, supply-chain and attribute hyperedges.

Metrics. We adopt commonly used *Accuracy*, *F1-score* and *Recall* for classification as the evaluation metric.

4.3. Experimental results

The experimental results of all competing methods are shown in Table 4. Furthermore, for graph-based and hypergraph-based baselines, we provide additional experimental results of these methods with and without the KGEN module. For these baselines with KGEN, the representations of enterprise vertices are initialized with the pretrained KGEN used in our MS-CGNN. As seen from the table, MS-CGNN achieves the state-of-the-art performance on all the metrics. Specifically, MS-CGNN outperforms the previous best HGNN (i.e., *HGAT*) by a large margin, increasing the *F1-score*, *accuracy*, and *recall* by 18.42%, 5.19% and 38.68%, respectively. Among the baselines, the SML based methods performs the worst. This is because these methods only represent enterprises with credit-related indicators and overlook the relationships among enterprises. However, not all the indicators of enterprises are available in many cases, which prevents these methods from effectively characterizing enterprises and results in low performance results. By incorporating pairwise relationships between enterprises, the GNN based methods obtains a big performance improvement, e.g., a 11.34% increase in accuracy.

HGAT improves enterprise representations by the topology structures of all three hypergraphs, which takes into account high-order relationships among enterprises that has been demonstrated useful for representation learning (Feng et al., 2019). Unfortunately, *HGAT* only focus on high-order topology structures and thus cannot capture multi-level semantics of different granularity. This makes the enterprise representations learned by *HGAT* cannot effectively characterize the risk features of enterprises, which impedes the performance improvement of ECR assessment. By contrast, our MS-CGNN can learn multi-level graph convolutional features based on both pairwise and high-order topology structure information. These structures at different levels of granularity carry complementary semantic information, which is useful for learning richer enterprise representations and thus improves the prediction results.

From Table 4, we also observe that all baselines obtain a performance gain using the KGEN on all the metrics. However, our MS-CGNN still outperforms competing baseline methods by a large margin, achieving at least 18.42% improvement on F1. This demonstrates the effectiveness of our method on the task of ECR assessment.

Besides, we also compare the computation budget and model parameters of different GNN-based methods and show the results in Table 5. From the table, we can see that the running time of Hypergra ph-based models (i.e., MS-CGNN and HyperGAT) are longer than graph-based models (i.e., RGCN, GraphSAGE and GAT) that use pairwise enterprise relationships. The training of MS-CGNN cost more time than that of HyperGAT. This is because MS-CGNN uses two HyperGAT in MHAN (module II) to encode homogeneous equity and supply-chain

[3] http://www.tianyancha.com

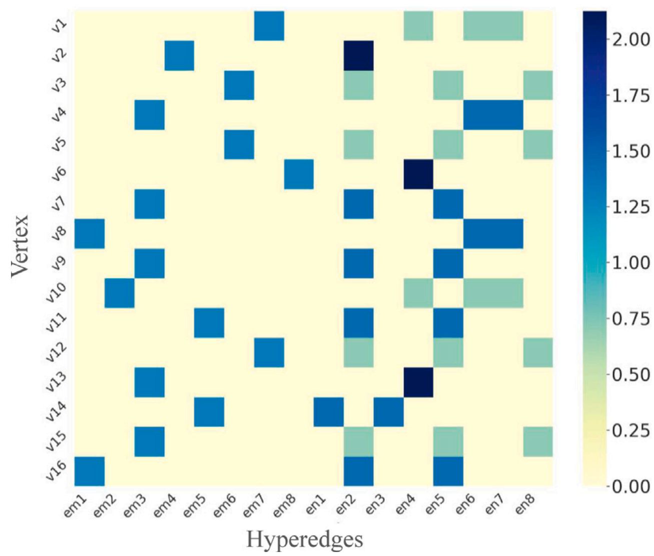

Fig. 9. An example of heatmap that corresponds to a local part of the learned hyperedge weight matrix.

hypergraphs, and one more HyperGAT-like structure in HDAN (module III) to encode heterogeneous hypergraphs.

For GPU memory and parameter number, MS-CGNN achieves comparable results with HyperGAT. This is because the memory cost and parameter number of hypergraph attention-based methods are closely related to the learning of hypergraph attention (Bai et al., 2021). Although two HyperGAT are used in Module II of our MS-CGNN, the sizes of vertex features and hyperedge features used in MS-CGNN are half the sizes of those features used in HyperGAT. This significantly reduces the number of parameters and memory usage of the HyperGAT used in our Module II. Finally, the number of parameters and memory usage of our Module II is similar to that of the baseline HyperGAT. As Module III only uses one head for learning hypergraph attention, it does not bring much parameters, which makes our MS-CGNN and *HyperGAT+KGEN* have similar parameter number. Since MS-CGNN needs to maintain additional weight matrices for each type of hyperedge, it requires slightly more GPU memory (about 176 MB) than the conventional HyperGAT.

To illustrate the new hyperedge weight matrix, we show an example of heatmap that corresponds to a local part of the learned hyperedge weight matrix. Specifically, we select 16 vertices and 16 hyperedges of two different types to form a local hyperedge weight matrix for display, and show its corresponding heatmap in Fig. 9. Specifically, in the figure $em1 \sim em8$ denote the *industry* hyperedge, while $en1 \sim en8$ denote the *performance_status* hyperedge. The depth of the cell color represents the influence of the hyperedges on the representation learning of vertices. We can observe that cells related to hyperedges of the *industry* type show the same color, whereas cells related to hyperedges of the *performance_status* type show different colors. This is because the number of the *industry* hyperedge connected to each vertex of $v1 \sim v16$ is the same. In contrast, according to Eq. (10) of our manuscript, the number of the *performance_status* hyperedge connected to each vertex are not the same, which leads to one hyperedge having different weights for different vertices. This is reasonable because one hyperedge may have a different influence on different vertices to which it is connected. The reason lies that the vertices of the hyperedge may connect to different numbers of hyperedges belonging to the same type, and the hyperedges of the same type should have the same influence on vertices.

Table 6
The results of the ablation study.

Model component	Recall	F1-score	Accuracy
MS-CGNN (full model)	0.9300	0.8863	0.9442
-w/o Weight	0.5802	0.7085	0.8885
-w/o Equity	0.4938	0.6593	0.8808
-w/o Supply-chain	0.4074	0.5773	0.8606
-w/o KGEN	0.8385	0.8360	0.8817
HDAN only	0.9259	0.7463	0.8529
HDAN w/o Weight	0.0494	0.0941	0.7779

4.4. Ablation study

In this part, we test the performance of MS-CGNN variants with different configurations, including: (i) *MS-CGNN w/o Weight*: it is obtained by replacing the redefined weight matrices for each hyperedge type in Module III (i.e., *HDAN*) with an identity matrix. (ii) *HDAN only*: it is obtained by removing Module I and II, and only use the *HDAN* module to learn enterprise representations based on the attribute hypergraph. (iii) *HDAN w/o Weight*: it is obtained by removing Module I and II, as well as replacing the redefined hyperedge weight matrix of HDAN with an identity matrix. Like variant *HDAN only*, this variant also only uses the attribute hypergraph to learn enterprise representations. (iv) *MS-CGNN w/o Equity*: it is obtained by removing the equity hypergraph of Module II. (v) *MS-CGNN w/o Supply-chain*: it is obtained by removing the supply-chain hypergraph of Module II. (vi) *MS-CGNN w/o KGEN*: it is obtained by removing Module I, i.e., KGEN.

The experimental results of the variants are reported in Table 6. As seen from the table, without redefining the weight matrices for each hyperedge type, the performance of variant *MS-CGNN w/o Weight* shows a big drop, e.g., 17.78% on *F1-score*, which demonstrates the effectiveness of the hyperedge-type dependent attention in Module III. This is because this variant only perform hypergraph convolutions with conventional attention mechanisms, which treat the hyperedges of different types equally and cannot explicitly distinguishing the effects of different types of hyperedges. This would result in illogical hyperedge weights, e.g., the same weights for hyperedges of different types, or significant different weights for hyperedges of the same type, resulting in performance degradation. The effectiveness of the redefined hyperedge weight matrices can also be demonstrated by the comparisons between variants *HDAN only* and *HDAN w/o Weight*. We can observe that *HDAN w/o Weight* suffers from a drastic drop and its performance is the worst among all the variants. This is because *HDAN w/o Weight* only exploits the attribute hypergraph for learning enterprise representations, and cannot distinguish the influences of hyperedges of different types. Without Module I and II, *HDAN only* witnesses a performance drop on all three metrics, e.g., 14.0% on *F1-score* and 9.13% on *Accuracy*. This demonstrates the necessity of learning enterprise representations from multi-level graph structures of different granularity, which carry complementary semantic information that helps to enrich enterprise representations. We also test the influences of high-order equity and supply-chain relationships by comparing the full model with *MS-CGNN w/o Equity* and *MS-CGNN w/o Supply-chain*, respectively. We can see that the performance of these two variants has a large drop on all the metrics, which indicates the effectiveness of encoding these two high-order graph topology structures into enterprise representations. The performance of *MS-CGNN w/o Equity* is better than that of *MS-CGNN w/o Supply-chain*, the main reason is that the structure of equity hypergraph is much sparser than that of supply-chain hypergraph, which is not conducive to learning rich enterprise representations. Specifically, the number of isolated vertices in the supply-chain and equity hypergraphs are 1590 and 2468, respectively.

After removing Module KGEN, the performance of our MS-CGNN witnesses a significant performance drop on all metrics. This indicates the effectiveness of the KGEN that encodes the graph structure of pairwise enterprise relationships. Furthermore, the importance of incorporating homogeneous high-order relations used in MHAN can be demonstrated by comparing variants *MS-CGNN w/o KGEN* and *HDAN only*. Specifically, the variant using both HDAN and MHAN (i.e., *MS-CGNN w/o KGEN*) outperforms the variant using only HDAN by a large margin, e.g., 8.97% in terms of F1-score. The reasons mainly lie in that variant *HDAN only* only focuses on high-order interactions derived from non-financial enterprise attributes, which overlooks pairwise and high-order financial interactions among enterprises that have a large effect on enterprise credit risk (Zhu, Zhou, Xie, et al., 2019), and thus impedes the risk prediction performance.

5. Conclusions

In this paper, we propose a new MS-CGNN framework for ECR assessment, which simultaneously takes into account multiple graph structures of different granularity for enterprise representation learning. The employment of topology structures of these graphs effectively alleviate the deficiency problem of financial indicators in enterprise representation learning. The structures from pairwise and high-order graphs of different granularity carry complementary information, which helps to enrich enterprise representations. Besides, to distinguish the influences of different types of hyperedges, we design a new hyperedge-type dependent attention mechanism for heterogeneous hypergraph convolutions. Experimental results on real-world ECR assessment datasets demonstrate the effectiveness of our MS-CGNN.

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

Data will be made available on request.

Acknowledgments

The research was supported in part by the National Key Research and Development Program of China under Grant 2020AAA0108500, National Nature Science Foundation of China under Grant Nos. 6210 2321, Fundamental Research Funds for the Central Universities, China under Grant No. D5000230095, D5000200146, and Shaanxi Province Key Industrial Project under Grant No. 2021ZDLGY03-08.

References

Altman, E. I., Iwanicz-Drozdowska, M., Laitinen, E. K., & Suvas, A. (2017). Financial distress prediction in an international context: A review and empirical analysis of Altman's Z-score model. *Journal of International Financial Management & Accounting*, *28*(2), 131–171.

Arora, N., & Kaur, P. D. (2020). A bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment. *Applied Soft Computing*, *86*, Article 105936.

Bacciu, D., Errica, F., Micheli, A., & Podda, M. (2020). A gentle introduction to deep learning for graphs. *Neural Networks*, *129*, 203–221.

Bai, S., Zhang, F., & Torr, P. H. (2021). Hypergraph convolution and hypergraph attention. *Pattern Recognition*, *110*, Article 107637.

Chen, C., Cheng, Z., Li, Z., & Wang, M. (2020). Hypergraph attention networks. In *2020 IEEE 19th international conference on trust, security and privacy in computing and communications* (pp. 1560–1565).

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., et al. (2015). Xgboost: Extreme gradient boosting. *R package version 0.4-2*, *1*(4), 1–4.

Cheng, D., Zhang, Y., Yang, F., Tu, Y., Niu, Z., & Zhang, L. (2019). A dynamic default prediction framework for networked-guarantee loans. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 2547–2555).

Derelioğlu, G., Gürgen, F., & Okay, N. (2009). A neural approach for sme's credit risk analysis in Turkey. In *Machine learning and data mining in pattern recognition: 6th International conference, MLDM 2009, Leipzig, Germany, July 23-25, 2009. Proceedings 6* (pp. 749–759).

Fang, K., Wu, J., Zhu, J., & Xie, B. (2010). Forecasting of credit card credit risk under asymmetric information based on nonparametric random forests. *Economic Research Journal*, *39*, 97–107.

Feng, Y., You, H., Zhang, Z., Ji, R., & Gao, Y. (2019). Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 33*, (no. 01), (pp. 3558–3565).

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, *30*.

Han, S. (2020). Semi-supervised learning classification based on generalized additive logistic regression for corporate credit anomaly detection. *IEEE Access*, *8*, 199060–199069.

Hu, X., Hu, J., Chen, L., & Li, Y. (2021). Credit risk assessment model for small, medium and micro enterprises based on RS-PSO-svm integration. In *IEEE 6th international conference on cloud computing and big data analytics* (pp. 342–345).

Huang, J., Huang, X., & Yang, J. (2021). Residual enhanced multi-hypergraph neural network. In *IEEE international conference on image processing* (pp. 3657–3661).

Jiang, B., Chen, Y., Wang, B., Xu, H., & Luo, B. (2023). DropAGG: Robust graph neural networks via drop aggregation. *Neural Networks*, *163*, 65–74.

Lengellé, R., & Denoeux, T. (1996). Training MLPs layer by layer using an objective function for internal representations. *Neural Networks*, *9*(1), 83–97.

Li, M., Ma, Z., Wang, Y. G., & Zhuang, X. (2020). Fast Haar transforms for graph neural networks. *Neural Networks*, *128*, 188–198.

Li, J., & Zhu, Q. (2019). Semi-supervised self-training method based on an optimum-path forest. *IEEE Access*, *7*, 36388–36399.

Liu, J., Song, L., Wang, G., & Shang, X. (2023). Meta-HGT: Metapath-aware HyperGraph transformer for heterogeneous information network embedding. *Neural Networks*, *157*, 65–76.

Phan, A. V., Nguyen, M. L., Nguyen, Y. L. H., & Bui, L. T. (2018). DGCNN: A convolutional neural network over large-scale labeled graphs. *Neural Networks*, *108*, 533–543.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Berg, R. v. d., Titov, I., & Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European semantic web conference* (pp. 593–607). Springer.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Sun, X., Yin, H., Liu, B., Chen, H., Cao, J., Shao, Y., et al. (2021). Heterogeneous hypergraph embedding for graph classification. In *Proceedings of the 14th acm international conference on web search and data mining* (pp. 725–733).

Tang, J., Li, J., Xu, W., Tian, Y., Ju, X., & Zhang, J. (2021). Robust cost-sensitive kernel method with Blinex loss and its applications in credit risk evaluation. *Neural Networks*, *143*, 327–344.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.

Wang, J. (2021). A study on credit risk assessment and credit decision making based on micro, small and medium enterprises. *Journal of Physics: Conference Series*, *1903*(1), Article 012037.

Wang, C., Yu, F., Zhang, Z., & Zhang, J. (2021). Multiview graph learning for small- and medium-sized enterprises' credit risk assessment in supply chain finance. *Complexity*, *2021*.

Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 28*, (no. 1).

Wang, D., Zhang, Z., Zhou, J., Cui, P., Fang, J., Jia, Q., et al. (2021). Temporal-aware graph neural network for credit risk prediction. In *Proceedings of the SIAM international conference on data mining* (pp. 702–710).

Xiong, X., Jia, M., Wenjie, Z., et al. (2009). Credit risk evaluation under the supply chain finance model. *Nankai Management Review*, *4*, 92–98.

Yang, S., Zhang, Z., Zhou, J., Wang, Y., Sun, W., Zhong, X., et al. (2021). Financial risk analysis for SMEs with graph-based supply chain mining. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence* (pp. 4661–4667).

Yao, G., Hu, X., & Wang, G. (2022). A novel ensemble feature selection method by integrating multiple ranking information combined with an SVM ensemble model for enterprise credit risk prediction in the supply chain. *Expert Systems with Applications*, *200*, Article 117002.

Zhang, L., Hu, H., & Zhang, D. (2015). A credit risk assessment model based on SVM for small and medium enterprises in supply chain finance. *Financial Innovation*, *1*(1), 1–21.

Zhang, Y., Wang, D., Chen, Y., Shang, H., & Tian, Q. (2017). Credit risk assessment based on long short-term memory model. In *Proceedings of the 13th international conference on intelligent computing theories and application* (pp. 700–712).

Zhang, J., & Zhang, B. (2016). The application of generalized semi-parametric additive credit score model based on group-LASSO method. *Journal of Applied Statistics and Management*, *35*(3), 517–524.

Zhao, Z., Chen, D., Wang, L., & Han, C. (2018). Credit risk diffusion in supply chain finance: A complex networks perspective. *Sustainability*, *10*(12), 4608.

Zhu, J., Zhao, X., Hu, H., & Gao, Y. (2019). Emotion recognition from physiological signals using multi-hypergraph neural networks. In *2019 IEEE international conference on multimedia and expo* (pp. 610–615).

Zhu, Y., Zhou, L., Xie, C., Wang, G.-J., & Nguyen, T. V. (2019). Forecasting SMEs' credit risk in supply chain finance with an enhanced hybrid ensemble machine learning approach. *International Journal of Production Economics*, *211*, 22–33.