Enterprise Bankruptcy Prediction with Meta-Path Denoising and Capsule Network Modeling

Hongrui Guo[1], Boyuan Ren[1], Hongzhi Liu[1]✉, Tianqi Sun[1], and Zhonghai Wu[1,2]

[1] School of Software and Microelectronics, Peking University, Beijing, 100871, China
{hongruiguo,renby,suntq}.stu.pku.edu.cn,liuhz.pku.edu.cn
[2] National Engineering Center of Software Engineering, Peking University, Beijing, 100871, China
wuzh.pku.edu.cn

Abstract. Enterprise bankruptcy prediction is essential for financial investment and corporate management. Traditional machine learning based methods mainly focus on internal risk mining while ignoring the external risks propagated among enterprises. Recent research efforts have attempted to use graph neural networks to model external risks. However, these methods generally lack interpretability and suffer from transferability. To address these issues, we propose a novel enterprise bankruptcy prediction model with meta-path denoising and capsule network modeling. Specifically, we design an automatic meta-path generation and selection method to improve the model's transferability while minimizing the information loss and noise introduction. Furthermore, a hierarchical meta-path information aggregation method is designed to enhance the model's interpretability, and a capsule network based risk assessment module is designed to dynamically capture enterprises' risky and non-risky factors for better bankruptcy prediction. Extensive experiments demonstrate the effectiveness and interpretability of the proposed model.

Keywords: Enterprise bankruptcy prediction · Heterogeneous graph · Meta-path selection · Capsule network

1 Introduction

Enterprises serve as the cornerstone of market economies with small and medium-sized enterprises (SMEs) playing a pivotal role. SMEs collectively constitute a substantial segment of the national economy. For example, SMEs account for approximately 80% of employment, over 60% of GDP, and more than 50% of tax revenue in China. The identification and analysis of enterprise

Hongrui Guo and Boyuan Ren contribute equally to this work.

bankruptcy patterns are crucial not only for maintaining economic stability but also for informing investment decisions and enhancing enterprise management strategies.

Most of the existing research adopts statistical analysis and traditional machine learning methods to predict enterprise bankruptcy risk, focusing on internal risk mining from financial data [1, 2]. However, due to the lack of regular financial reports, especially for SMEs, the application of these methods faces significant challenges. In addition, these methods often ignore the external risks propagated among enterprises. Compared with large enterprises, SMEs are more vulnerable to external shocks and have weaker resilience.

In recent years, given the advantages of graph models in representing and capturing multidimensional relationships, some studies tried to explore their application in predicting enterprise bankruptcy, especially in simulating the propagation of external risks [3, 4]. Recent efforts have attempted to model external risks using heterogeneous graph, such as HAT [5] and ComRisk [6], but these methods generally lack interpretability and suffer from transferability.

To address these issues, we propose a novel framework to predict enterprise bankruptcy with **m**eta-path **d**enoising and **c**apsule **n**etwork modeling (called MDCN for short). Specifically, a meta-path based explicit risk modeling module is designed to capture the complex external association risks among enterprises. To enhance the model's transferability while avoiding the loss of important information and the introduction of noise, an automatic meta-path generation and selection method is designed to extract effective meta-paths. Furthermore, a hierarchical meta-path aggregation method is designed to capture the external association risks and improve the interpretability of the meta-paths. To enhance the expression ability of embedding representations for better bankruptcy prediction, a capsule network based risk assessment module is designed to dynamically extract both risky and non-risky factors of enterprises.

2 Related Work

Early related studies focused on statistical analysis [1]. Statistical techniques, such as linear discriminant analysis (LDA), multivariate discriminate analysis (MDA), quadratic discriminant analysis (QDA), logistic regression (LR) and factor analysis (FA), have been widely used for risk analysis and bankruptcy prediction in practice [2]. With advancements in machine learning, methods such as support vector machines [7, 8], multi-layer perception (MLP) [9] and decision trees [10, 11] have gained popularity. These methods focused on mining and making use of effective financial ratios (variables), such cash flow, net income and debt cost. However, they always treat enterprises as isolated nodes, ignoring their external association risks which are crucial as they can provide additional insights, such as the impact of market conditions, industry trends, and so on.

Given the advantages of graph models in representing and modeling graph-structured data, several recent studies tried to construct enterprise relationship graph and design corresponding models for bankruptcy prediction and fraud detection. HAT [5] constructs meta-paths in enterprise relationship graphs to capture intricate semantic relationships, addressing node heterogeneity. HIDAM [12] employs a multi-layer attention mechanism to evaluate intra- and inter-meta-path relationships, enhancing bankruptcy prediction accuracy for SMEs. SemiGNN [13] uses multi-view data and proposes an attention mechanism to enhance the relationships among various neighbors and perspectives for fraud detection. FraudCom [14] employs a competitive graph neural network as its foundational component, modeling the distributions of normal and fraudulent behaviors separately to enhance fraud detection.

Despite the success of graph based models, challenges remain in modeling external enterprise risks. Many models struggle to accurately assess which associations have the most significant impact on enterprise risk evaluation due to a lack of interpretability. While some meta-path based models offer improved interpretability, they still encounter challenges such as limited transferability and susceptibility to noise interference. Specifically, methods based on meta-paths can generally be divided into two categories: the former includes models that rely on manually preset meta-paths [15, 16], which limits the transferability of the module due to their heavy dependence on the original data. The later consists of models that utilize all meta-paths not exceeding a fixed length [17, 18], which introduces significant irrelevant noise and reduces the accuracy of information aggregation.

3 Problem Definition

Let $C = \{c_1, c_2, ..., c_N\}$ denote the set of enterprises, where N is the total number of enterprises. For each enterprise $c_i \in C$, we assume some internal data is known, denoted as Int_i. It may include some basic attributes, such as establishment time and registered capital, as well as litigation data, such as the cause and result of litigation.

In addition, we assume there are some external association information for each enterprise c_i, denoted as Ext_i. It may include interactions with suppliers, customers, broader market, and so on. External risks like market fluctuations, consumer behavior changes, and supply chain disruptions can greatly affect cash flow, profitability and solvency. Thus, integrating these external factors into a bankruptcy assessment model is vital for improving its accuracy.

This paper aims to model both internal and external risks to predict the likelihood of enterprise bankruptcy. Specifically, our goal is to learn a prediction function $f(\cdot)$ based on enterprises' internal information and external association information. It can be defined as: $f(Int_i, Ext_i) = p(y_i = 1|Int_i, Ext_i)$ i.e. predicting the probability that enterprise c_i will bankrupt ($y_i = 1$) given its internal information Int_i and external association information Ext_i.

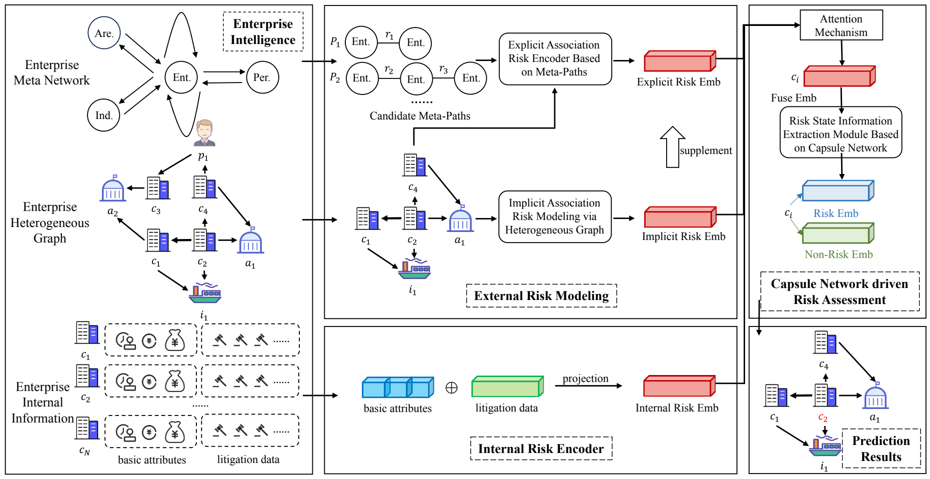

Fig. 1: Architecture of the proposed MDCN model

4 The Proposed Method

The architecture of the proposed model is illustrated in Fig. 1. Besides a traditional internal risk encoder, it contains two novel modules, i.e. an external risk modeling module and a risk assessment module. Specifically, to better model the external risk, we design an explicit association risk encoder based on meta-paths and an implicit association risk modeling component via heterogeneous graph. The former focuses on generating and filtering out invalid paths to retain essential association information for bankruptcy prediction. Simultaneously, it hierarchically aggregates external risk information to nodes, enhancing the interpretability of risk propagation. The latter focuses on capturing global implicit association information to supplement the lost information caused by limited length of meta-paths. Furthermore, to better fuse and make use of different risk information for accurate bankruptcy prediction, we design a capsule network driven risk assessment method.

4.1 Explicit Association Risk Encoder Based on Meta-Paths

With the given data about enterprises and related objects, we can construct a heterogeneous graph $G = \{\mathcal{V}, \mathcal{E}\}$ and a corresponding meta-network $T_G = \{\mathcal{A}, \mathcal{R}\}$ as shown in the upper left corner of Fig. 1. The network schema T_G is a meta-template for the heterogeneous graph G with an object type mapping $\phi : \mathcal{V} \to \mathcal{A}$ and a link mapping $\psi : \mathcal{E} \to \mathcal{R}$. Each object $v \in \mathcal{V}$ belongs to a particular object type $\phi(v) \in \mathcal{A}$, such as 'Enterprise', 'Person', 'Industry' and 'Area'. Each link $e \in \mathcal{E}$ belongs to a particular relation $\psi(e) \in \mathcal{R}$, such as 'Invest', 'Branch', 'Belong' and 'Contain'. With the help of the meta network T_G, we can construct semantic meta-paths between objects, such as $Enterprise \overset{Belong}{\to}$

$Industry \overset{Contain}{\rightarrow} Enterprise$ represents the enterprises in the same industry. Furthermore, we can utilize meta-paths to extract external associations between enterprise nodes and capture the external risks of enterprises.

Meta-Path Denoising. Given the meta network T_G, we can generate a large number of meta-paths. However, most of them are irrelevant to the given prediction task. To balance the transferability while reducing the interference of irrelevant information, we designed an automatic meta-path selection method. Specifically, with consideration of both the computational cost and the given prediction task, we only consider the meta-paths with both end nodes are *Enterprise* and length not exceeding a given threshold l. We assume that an effective meta-path should predominantly provide either risk or non-risk information, with a focus on one type. Furthermore, if the proportions of various information are too similar, the meta-path's function may become ambiguous. The effectiveness of a given meta-path P is calculated as follows:

$$S_P = \max\left(\frac{N_P^{rr}}{T_P} \times \frac{N_P^{rr}}{N_P^{nn}+1}, \frac{N_P^{nn}}{T_P} \times \frac{N_P^{nn}}{N_P^{rr}+1}\right) \tag{1}$$

where $T_P = N_P^{rr} + N_P^{nr} + N_P^{rn} + N_P^{nn}$ denotes the total number of meta-path P's instances. $N_P^{rr}(N_P^{nn})$ represents the frequency of P's instances where both end nodes are risky(non-risky) enterprises. N_P^{nr} (N_P^{rn}) represents the frequency of P's instances with non-risky (risky) target enterprise and risky (non-risky) source enterprise. The penalty terms $\frac{N_P^{rr}}{N_P^{nn}+1}$ and $\frac{N_P^{nn}}{N_P^{rr}+1}$ are used to reduce the priority of meta-paths whose ratios of various information are too similar. With the effectiveness measure of meta-paths, the *top-k* ranked candidate meta-paths are chosen for the following module to capture explicit association risk.

To accurately model the external risk characteristics of enterprises, we design a hierarchical attention structure, as shown in Fig. 2. It consists of two levels, i.e. the instance level and the semantic level. The instance level focuses on aggregating information along meta-path instances to compute semantic embeddings for each node across different meta-paths. The semantic level further considers the semantic differences among different meta-paths and integrates their embeddings to derive the external risk embedding along explicit paths.

Instance-Level Fusion. For a target enterprise node c_i and a given meta-path P, there are many different path instances. We assume the importance of these path instances for the target enterprise c_i are different, which is depended on the source node c_k. Furthermore, we design a dynamic attention layer to perform weighted aggregation on the path instances of meta-path P for target node c_i as the following:

$$Z_i^P = h_i^{mt} + W_P \sum_{c_k \in N_i^P} \alpha_{p_{ik}} \cdot h_{p_{ik}} \tag{2}$$

where $Z_i^P \in \mathbb{R}^{d_o}$ denotes the enterprise c_i's semantic embedding representation under meta-path P. $h_i^{mt} \in \mathbb{R}^{d_o}$ and $h_k^{mt} \in \mathbb{R}^{d_o}$ denote the current embedding of enterprise c_i and c_k, respectively. $h_{p_{ik}} = [h_i^{mt}||h_k^{mt}]$ represents the embedding of

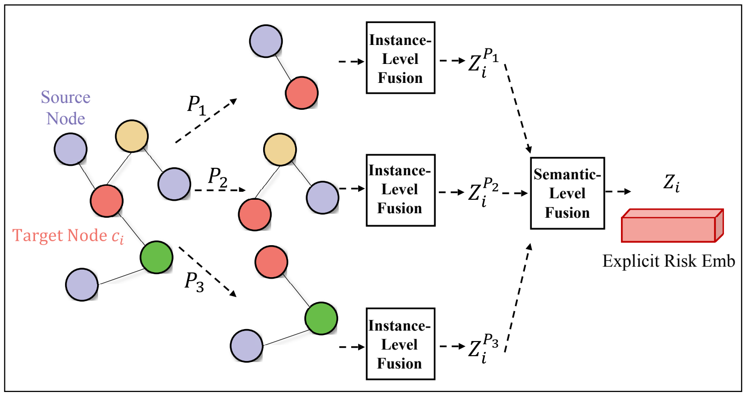

Fig. 2: Meta-path hierarchical aggregation

the path instance p_{ik} where c_i is the target node and c_k is the source node, and $||$ denotes the concatenation operation. The learnable parameter matrix associated with meta-path P is $W_P \in \mathbb{R}^{2d_o \times d_o}$ which is used for information extraction from the fused path instance embedding. $\alpha_{p_{ik}}$ denotes the attention weight of instance path p_{ik} and is calculated as follows:

$$\alpha_{p_{ik}} = \frac{\exp\left(\sigma\left(W_a^P h_{p_{ik}}\right)\right)}{\sum_{c_j \in N_i^P} \exp\left(\sigma\left(W_a^P h_{p_{ij}}\right)\right)} \tag{3}$$

where N_i^P denotes the meta-path neighbors of enterprise c_i under the meta-path P. The learnable parameter vector corresponding to meta-path P is denoted as $W_a^P \in \mathbb{R}^{2d_o \times 1}$ which is used to map path instance embeddings to fusion weights. $\sigma(.)$ represents the activation function, such as ReLU [19].

Semantic-Level Fusion. To integrate the semantic information from different meta-paths for each enterprise node c_i, we propose to fuse the semantic representation Z_i^P with an attention mechanism as follows:

$$Z_i = \sum_{P \in T_i} \beta_i^P Z_i^p \tag{4}$$

where T_i is the set of meta-path related to the target node c_i. β_i^P denotes the attention weights, which is calculated as follows:

$$\beta_i^P = \frac{\exp\left(\sigma\left(W_b Z_i^P\right)\right)}{\sum_{P' \in T_i} \exp\left(\sigma\left(W_b Z_i^{P'}\right)\right)} \tag{5}$$

where $W_b \in \mathbb{R}^{d_o \times 1}$ is a globally learnable meta-path weight vector which is used to map semantic embeddings to fusion weights.

4.2 Implicit Association Risk Modeling via Heterogeneous Graph

Due to the limitation of meta-path length, some associations between enterprises may be underutilized. To address this, we propose an implicit association risk modeling component via heterogeneous graph, which includes a relation-aware heterogeneous graph convolution network and a global attention based implicit association modeling module. The former aims to capture the risk pattern of enterprises, while the latter explores implicit associations among enterprises with similar patterns from a global perspective to enhance bankruptcy prediction.

Relation-aware heterogeneous graph convolution network. We aggregate neighbor information under different relationship types for each node $v_i \in \mathcal{V}$. Given the relation $\pi \in \mathcal{R}$, the aggregation operation is as follows:

$$P_i^\pi = \sum_{v_k \in N_i^\pi} \delta_{ik}^\pi \cdot h_k^{ht} \tag{6}$$

where $P_i^\pi \in \mathbb{R}^{d_o}$ denotes the neighbor-aggregated embedding of node v_i. N_i^π denotes the neighbors of node v_i. $h_k^{ht} \in \mathbb{R}^{d_o}$ represents the current embedding of neighbor node v_k. δ_{ik}^π denotes the aggregated weight of neighbor v_k and is calculated as follows:

$$\delta_{ik}^\pi = \frac{\exp\left(\sigma\left(W^\pi w_{ik}\right)\right)}{\sum_{v_j \in N_i^\pi} \exp\left(\sigma\left(W^\pi w_{ij}\right)\right)}, \qquad w_{ik} = \left[h_i^{ht} \| h_k^{ht}\right] \tag{7}$$

where $W^\pi \in \mathbb{R}^{2d_o \times 1}$ is the learnable weight vector under relationship π, which is used to obtain the attention fusion weights of the neighbors. Then we integrate the outputs under various relationships as follows:

$$P_i = \theta \cdot \sigma\left(h_i^{ht}\right) + \sum_{\pi \in \mathcal{R}} \eta_i^\pi \cdot \left(W_G P_i^\pi\right) \tag{8}$$

where $P_i \in \mathbb{R}^{d_o}$ denotes the current output embedding of node v_i. $W_G \in \mathbb{R}^{d_o \times d_o}$ is a globally learnable parameter matrix, which is used to map various results into the same feature space. θ is a learnable parameter that balances the importance of aggregated risk information and its own information. η_i^π denotes the fusion weight of relation π and is calculated as follow:

$$\eta_i^\pi = \mathrm{Softmax}_{\pi \in \mathcal{R}}\left(\frac{k_i^{\pi T} q_i^\pi}{\sqrt{d_o}}\right), \quad q_i^\pi = W_Q^\pi h_i^{mt} + b_Q^\pi, \quad k_i^\pi = W_K^\pi P_i^\pi + b_K^\pi \tag{9}$$

where $W_Q^\pi, W_K^\pi \in \mathbb{R}^{d_o \times d_o}$ and $b_Q^\pi, b_K^\pi \in \mathbb{R}^{d_o}$ are learnable parameters used to convert various embeddings into a uniform feature space to calculate similarity weight. Through L layers of heterogeneous graph convolution, we obtain the output embeddings $P^{(L)} \in \mathbb{R}^{|\mathcal{V}| \times d_o}$ for all nodes. Next, we extract enterprises' risk pattern embeddings $P_e^{(L)} \in \mathbb{R}^{N \times d_o}$ from $P^{(L)}$.

Global attention-based implicit association modeling. We integrate information between enterprises with similar risk patterns from a global perspective as follows:

$$\hat{Z}^{at} = \eta^{at}(W_V^{at} P_e^{(L)}) \tag{10}$$

where $\hat{Z}^{at} \in \mathbb{R}^{N \times d_o}$ is the enterprise embedding matrix obtained by attention. W_V^{at} denotes the learnable parameters used to obtain values in the attention mechanism. η^{at} denotes an attention weight matrix and is calculated like [20]:

$$\eta^{at} = \text{Attention}(P_e^{(L)}; W_Q^{at}, W_K^{at}, b_Q^{at}, b_K^{at}) \tag{11}$$

where $W_Q^{at}, W_K^{at} \in \mathbb{R}^{d_o \times d_o}$ and $b_Q^{at}, b_K^{at} \in \mathbb{R}^{d_o}$ are learnable parameters in the attention mechanism.

Then the final implicit risk embedding matrix of the enterprise $\hat{Z} \in \mathbb{R}^{N \times d_o}$ is calculated through a feedforward network(FFN) used to enhance feature representations through deep nonlinear mappings:

$$\hat{Z} = \text{FFN}(\hat{Z}^{at}) + \gamma P_e^{(L)} \tag{12}$$

where γ is a learnable parameter to balance the local and global risk information, and $\hat{Z}_i \in \hat{Z}$ represents the implicit risk embedding of enterprise node c_i.

4.3 Capsule Network driven Risk Assessment

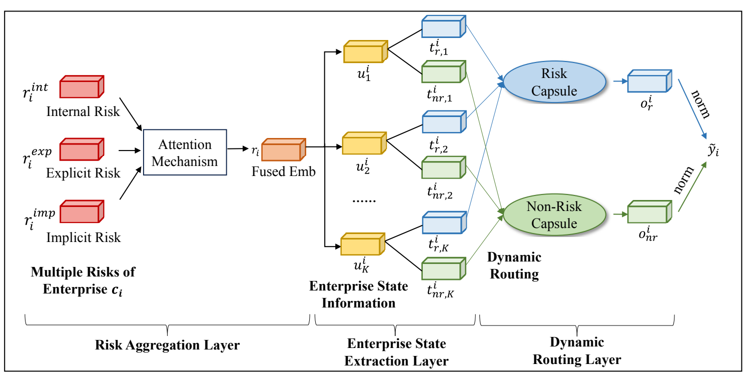

Fig. 3: Risk assessment with capsule network

To more comprehensively integrate various risk information and enhance the expressiveness of enterprise embeddings for bankruptcy prediction, a risk assessment module based on capsule networks is designed. It includes a risk aggregation layer, an enterprise state extraction layer, and a dynamic routing layer, as shown in Fig. 3. The functions of these parts are introduced in detail below.

Risk Aggregation Layer. To comprehensively consider various risk information, it is necessary to aggregate multiple types of risks. Besides the explicit external risk Z_i and the implicit external risk $\hat{Z}_i$, we model the internal risk of each enterprise c_i as [6] to generate the internal risk embedding $h_i \in \mathbb{R}^d$, which is also used as the initial embedding for the enterprise node. For other types of nodes in the heterogeneous graph, we directly use the pre-trained embeddings obtained through metapath2vec [21] as the initial embeddings. After obtaining internal and external risk embeddings, we map them into the same space. For example, for enterprise c_i :

$$r_i^{int} = W^{int} h_i, \quad r_i^{exp} = W^{exp} Z_i, \quad r_i^{imp} = W^{imp} \hat{Z}_i \tag{13}$$

where $r_i^{int}, r_i^{exp}, r_i^{imp} \in \mathbb{R}^{d_o}$ are the transformed risk embeddings. $W^{int} \in \mathbb{R}^{d \times d_o}, W^{exp} \in \mathbb{R}^{d_o \times d_o}, W^{imp} \in \mathbb{R}^{d_o \times d_o}$ are learnable transformation matrices. Then, the internal and external risks are merged as follows:

$$r_i = \sum_{t \in T^r} \alpha_i^t \cdot r_i^t, \qquad \alpha_i^t = \frac{\exp\left(W^t r_i^t\right)}{\sum_{k \in T^r} \exp\left(W^k r_i^k\right)} \tag{14}$$

where $r_i \in \mathbb{R}^{d_o}$ is the fused embedding, $T^r = \{int, exp, imp\}$ is the set of risk types, r_i^t is the risk embedding of type t, α_i^t is the weight of type t risk embedding, and $W^t \in \mathbb{R}^{d_o \times 1}$ is the learnable vector of type t risk.

Enterprise State Extraction Layer. This layer is used to further extract the enterprise state information from the merged embeddings. A simple linear layer is used to extract cap_num enterprise state information embeddings from the merged embedding of each enterprise. We calculate the k-th state information of enterprise c_i as follows:

$$u_k^i = W_k^{cap} r_i + b_k^{cap} \tag{15}$$

where $u_k^i \in \mathbb{R}^{d_o}$ is the k-th state information embedding, and $W_k^{cap} \in \mathbb{R}^{d_o \times d_o}$, $b_k^{cap} \in \mathbb{R}^{d_o}$ are the learnable parameters corresponding to the k-th state.

Dynamic Routing Layer. To enhance the expressiveness of the embedding representations, we propose to dynamically generate the risky embedding and non-risky embedding of each enterprise based on the enterprise's state information. To achieve these goals, a capsule architecture is designed. Specifically, two state capsules are used, namely the risky state capsule and the non-risky state capsule, which jointly select some state information and parse their semantics. Specifically, in each state capsule, the latent state features are extracted for each state information, represented as $t_{s,k}^i = W_{s,k} \cdot u_k^i$, where $t_{s,k}^i$ denotes the feature of state s extracted from the state information u_k^i, $W_{s,k} \in \mathbb{R}^{d_o \times d_o}$ is a learnable weight matrix, $s \in S$ and $S = \{r, nr\}$ denotes the set of risk and non-risk states. Next, the capsule of state s takes all the feature vectors $\{t_{s,k}^i | k = 1, 2, \cdots, cap\_num\}$ as inputs and generates its output vector through an iterative dynamic routing process. Essentially, the output of each state capsule can be considered as a weighted sum of these feature vectors:

$$t_s^i = \sum_k w_{s,k}^i \cdot t_{s,k}^i \tag{16}$$

where $t_s^i \in \mathbb{R}^{d_o}$ is the embedding corresponding to the risk state s, and $w_{s,k}^i$ is the coupling coefficient computed through the dynamic routing algorithm, indicating the importance of state information u_k^i in state s. In addition, the capsule uses the length of its output to represent the probability of the concept it signifies occurring. Specifically, a squash function is used to compress t_s^i into a vector with a length constrained to the range $(0, 1)$:

$$o_s^i = \frac{\left\|t_s^i\right\|^2}{1 + \|t_s^i\|^2} \frac{t_s^i}{\|t_s^i\|} \tag{17}$$

where $\|\cdot\|$ represents the length of a vector. The length of vector o_s^i is used to represent the probability that enterprise c_i is in state s.

Finally, the predicted bankruptcy probability $\tilde{y}_i$ of enterprise c_i is generated as follows:

$$\tilde{y}_i = \frac{\exp\left(\left\|o_r^i\right\|\right)}{\exp\left(\|o_r^i\|\right) + \exp\left(\|o_{nr}^i\|\right)} \tag{18}$$

4.4 Parameter Learning

The bankruptcy prediction task is essentially a binary classification problem, where the objective is to infer the risk label y of enterprises. To effectively learn the model parameters, we utilize the cross entropy loss as the objective function:

$$L = -\frac{1}{|C_{\text{train}}|} \sum_{c_i \in C_{\text{train}}} y_i \log\left(\tilde{y}_i\right) + (1 - y_i) \log\left(1 - \tilde{y}_i\right) \tag{19}$$

where $C_{\text{train}} \subseteq C$ represents the set of enterprises in the training dataset, y_i is the true label and $\tilde{y}_i$ is the predicted bankruptcy probability of enterprise c_i. We optimize the objective function by Adam [22], which can dynamically adjust the learning rate during model training. To improve robustness of the trained model, techniques like weight decay and regularization are applied.

5 Experiments

5.1 Experimental Setup

Dataset To validate the effectiveness of the proposed model, we utilize a manually extracted dataset SMEsD [6] as the experimental data. It is a real-world dataset for enterprise bankruptcy prediction, containing detailed information on 3,976 small and medium-sized enterprises (SMEs) and related individuals in China from 2014 to 2021.

Evaluation Metrics The same as in previous studies [5, 6], we adopt Accuracy, Recall, F1 score, and AUC as the evaluation metrics. Accuracy reflects the proportion of correct predictions. Recall measures the ratio of true positives identified to all actual positives. F1 score balances precision and recall. AUC evaluates a model's ability to distinguish between positive and negative classes.

Baselines To construct a comprehensive comparison, we adopt several different types of baseline methods as follows:

- **Traditional Machine Learning based Methods: LR** [23] employs logistic regression for binary classification. **SVM** [24] utilizes support vector machines to classify data via an optimal separating hyperplane. **GBDT** [25] combines weak classifiers to handle nonlinear data more effectively.
- **Homogeneous Graph Network based Methods: GCN** [26] aggregates neighbor information for node classification. **GAT** [27] uses an attention mechanism to weight neighbors differently for node classification.
- **Hypergraph Network based Methods: HGNN** [28] learns node representations by capturing higher-order relationships and dependencies with hyperedges. **HWNN** [29] applies localized hypergraph convolution using wavelet-based polynomial approximations.
- **Heterogeneous Graph Network based Methods: RGCN** [30] extends convolution with weight matrices for different relation types. **HAN** [31] uses attention over meta-paths in heterogeneous graphs. **HAT** [5] predicts bankruptcy with a neighbor encoding layer and three attention layers.
- **Hybrid Method: ComRisk** [6] combines an internal risk encoder and an external risk encoder to assess the risk of enterprise bankruptcy.

Implementation Details The experiments used optimal parameters from original papers for baseline methods or fine-tuned them using a validation set. For MDCN, internal and external risk output dimensions were set to 16 and 12, respectively, with the top 11 meta-paths selected for explicit risk encoding and 3 layers of heterogeneous graph convolution for implicit risk encoding. The risk assessment module extracted 4 states per enterprise. Section 5.4 analyzes the impact of important parameters. Neural networks were trained for 500 epochs using the Adam optimizer [22].

5.2 Comparison Experiment

The purpose of this experiment is to evaluate the performance of the proposed model compared with state-of-the-art methods. Table 1 shows the performance of different methods on the benchmark dataset. Bold numbers indicate the best results and underlined numbers indicate the second-best results.

The results indicate that our model significantly outperforms all baseline methods according to various evaluation metrics, which confirms the effectiveness of this approach for bankruptcy prediction. Traditional machine learning

Table 1: Performance comparison of different methods

Models	**Accuracy**	**Recall**	**F1-Score**	**AUC**
LR	0.6667	0.7883	0.7539	0.6382
SVM	0.6857	<u>0.8274</u>	0.7732	0.6928
GBDT	0.6814	0.7785	0.7599	0.6972
GCN	0.6983	0.7980	0.7741	0.7409
GAT	0.6962	0.8046	0.7743	0.7386
HGNN	0.6793	0.8241	0.7690	0.6981
HWNN	0.6603	0.7915	0.7512	0.6656
RGCN	0.7447	0.8208	0.8064	0.8026
HAN	0.7025	0.7752	0.7715	0.7344
HAT	0.6934	0.7507	0.7528	0.7403
ComRisk	<u>0.7574</u>	0.8013	<u>0.8105</u>	<u>0.8111</u>
MDCN	**0.7595**	**0.8434**	**0.8213**	**0.8149**

based methods, such as LR and GBDT, which ignore the external risk factors, perform relative worse among the baseline methods. This confirms the importance of external risk modeling for bankruptcy prediction. Homogeneous Graph Network based Methods, such as GCN and GAT, treat heterogeneous relations as homogeneous. Hypergraph Network based Methods, such as HGNN, can capture higher-order relationships. However, when handling complex heterogeneous information with diverse entities and relationships, these methods may show worse performance compared to methods via heterogeneous graph. This highlights the importance of modeling heterogeneous information. Heterogeneous Graph Network methods, like HAN and HAT, are limited by the meta-path length, which can miss implicit relational information. Besides, treating all meta-paths equally may also introduce irrelevant noise, affecting prediction performance.

Compared with the second best baseline ComRisk, the performance of MDCN improvement stems from the following aspects: (1) Using meta-path based algorithms to model the explicit association risks between enterprises and reducing irrelevant noise information through meta-path filtering, which enhances the model's accuracy, (2) Utilizing heterogeneous graph based methods to supplement the information not captured by meta-paths from a global perspective, improving the completeness of information utilization, (3) Separating the risky and non-risky states with capsule networks, thereby enhancing the information expressiveness of the embeddings.

5.3 Ablation Study

To study the effectiveness of different modules within the proposed model, we conduct several ablation experiments. Four model variants were designed: (1) w/o Intra, excluding the internal risk encoder; (2) w/o Meta, omitting the meta-path-based explicit association risk encoder; (3) w/o HA, removing the implicit association risk encoder; (4) w/o Cap, replacing the capsule network based risk

Table 2: Results of ablation study

Variants	**Accuracy**	**F1-Score**	**AUC**
w/o Intra	0.6603	0.7356	0.7276
w/o Meta	0.6878	0.7712	0.7056
w/o HA	0.7149	0.8053	0.7088
w/o Cap	0.7278	0.7916	0.8081
MDCN	**0.7595**	**0.8213**	**0.8149**

assessment module with compressed risk embeddings. The results are shown in Table 2. It is obvious that each module contributes positively to the final results.

5.4 Hyperparameter Analysis

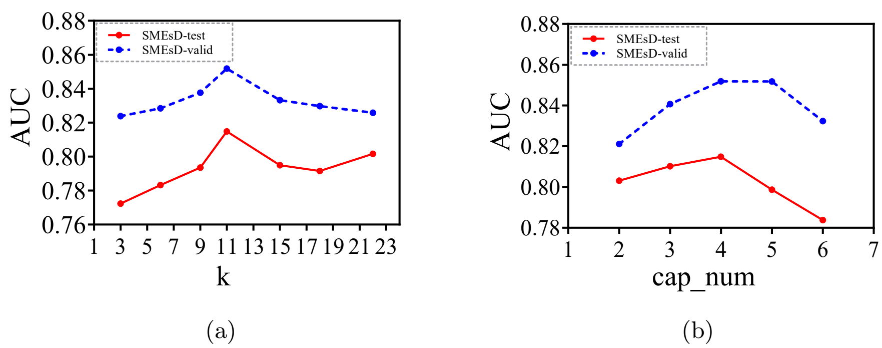

Fig. 4: Effects of different settings of k (the number of selected meta-paths) and cap_num (the number of enterprise state information extractions)

This experiment aims to evaluate the impact of various hyperparameter settings on the performance of the proposed model:

Fig. 4(a) shows that with increase in the number of selected meta-paths, the performance of the model first increases and then decreases. This confirms that the introduction of effective meta-paths can enhance the prediction ability of the model, while irrelevant paths may bring noise.

Fig. 4(b) shows the results obtained with different number of enterprise state information extractions. Fewer extractions may miss critical state information, whereas too many can introduce noise, reducing the accuracy of the risk embeddings generated during dynamic routing.

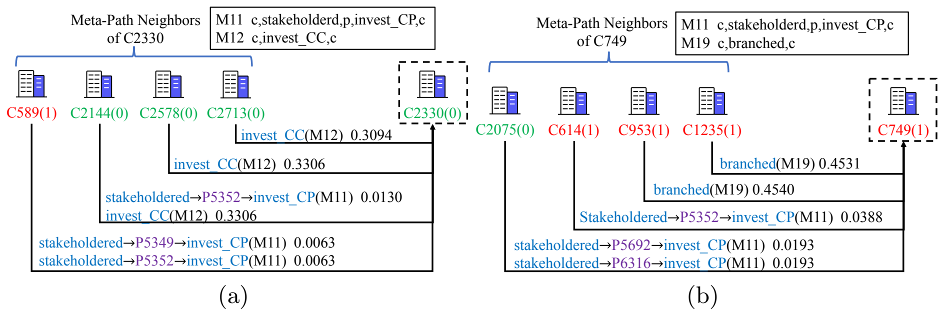

Fig. 5: Case study of non-risky enterprise C2330 and risky enterprise C749

5.5 Case Study

In addition to the excellent predictive ability, the explicit risk encoder based on meta-paths in our model also provides interpretability for the results. This section shows the inference of two representative enterprises from the test set: a non-risky enterprise C2330 and a risky enterprise C749. The meta-path instances with significant influence on them are shown in the Fig. 5 with the weights of path instances given on the edges.

Enterprise C2330: Fig. 5(a) shows the meta-path neighbors of enterprise C2330. MDCN can accurately identify C2330 as a non-risky enterprise. It can be observed that enterprises C2144, C2578, and C2713 are upstream investment enterprises of C2330, and these enterprises are non-risky. The meta-path instances (M12) with a higher weight provide sufficient non-risk information, supporting the correct identification of C2330 as a non-risky enterprise.

Enterprise C749: Fig. 5(b) shows the meta-path neighbors of enterprise C749. MDCN can accurately identify C749 as a risky enterprise. According to the meta-path instances and their corresponding attention weights, it can be seen that enterprises C953 and C1235 are subsidiaries of enterprise C749, and these subsidiaries have gone bankrupt. The high weights of their meta-path instances (M19) provide risk information for the assessment, helping MDCN to accurately identify C749 as a risky enterprise.

6 Conclusion

To address the shortcomings of existing enterprise bankruptcy prediction methods, this paper proposes a novel bankruptcy prediction framework. It boosts prediction by generating and selecting effective meta-paths to minimize the noise introduction, while employing a capsule network to capture both risky and non-risky factors, thereby improving embedding representations. Additionally, it increases interpretability through a hierarchical meta-path aggregation method that models external association risks. In this paper, we only make use of the internal attributes and external connection information, treating them as sectional data. In the future, we will integrate macroeconomic factors to enhance the model's adaptability to economic environments and introduce time-series data to improve its capacity to capture the dynamic risks faced by enterprises.

Acknowledgments. This study was funded by the Major Science and Technology Program of Changsha, China (Grant No. kh2401025) as well as the Sichuan Science and Technology Program (Grant No. 2024YFHZ0022).

References

1. Alaka, H.A., Oyedele, L.O., Owolabi, H.A., Kumar, V., Ajayi, S.O., Akinade, O.O., Bilal, M.: Systematic review of bankruptcy prediction models: Towards a framework for tool selection. Expert Systems with Applications **94**, 164–184 (2018)
2. Kumar, P.R., Ravi, V.: Bankruptcy prediction in banks and firms via statistical and intelligent techniques–a review. European Journal of Operational Research **180**(1), 1–28 (2007)
3. Cheng, D., Zhang, Y., Yang, F., Tu, Y., Niu, Z., Zhang, L.: 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 (2019)
4. Shumovskaia, V., Fedyanin, K., Sukharev, I., Berestnev, D., Panov, M.: Linking bank clients using graph neural networks powered by rich transactional data. International Journal of Data Science and Analytics **12**(2), 135–145 (2021)
5. Zheng, Y., Lee, V.C., Wu, Z., Pan, S.: Heterogeneous graph attention network for small and medium-sized enterprises bankruptcy prediction. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 140–151 (2021)
6. Wei, S., Lv, J., Guo, Y., Yang, Q., Chen, X., Zhao, Y., Li, Q., Zhuang, F., Kou, G.: Combining intra-risk and contagion risk for enterprise bankruptcy prediction using graph neural networks. Information Sciences **659**, 120081 (2024)
7. Min, J.H., Lee, Y.C.: Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. Expert Systems with Applications **28**(4), 603–614 (2005)
8. Huang, C.L., Chen, M.C., Wang, C.J.: Credit scoring with a data mining approach based on support vector machines. Expert Systems with Applications **33**(4), 847–856 (2007)
9. Atiya, A.F.: Bankruptcy prediction for credit risk using neural networks: A survey and new results. IEEE Transactions on Neural Networks **12**(4), 929–935 (2001)
10. Olson, D.L., Delen, D., Meng, Y.: Comparative analysis of data mining methods for bankruptcy prediction. Decision Support Systems **52**(2), 464–473 (2012)
11. Delen, D., Kuzey, C., Uyar, A.: Measuring firm performance using financial ratios: A decision tree approach. Expert Systems with Applications **40**(10), 3970–3983 (2013)
12. Zhang, Z., Ji, Y., Shen, J., Zhang, X., Yang, G.: Heterogeneous information network based default analysis on banking micro and small enterprise users. arXiv preprint arXiv:2204.11849 (2022)
13. Wang, D., Lin, J., Cui, P., Jia, Q., Wang, Z., Fang, Y., Yu, Q., Zhou, J., Yang, S., Qi, Y.: A semi-supervised graph attentive network for financial fraud detection. In: 2019 IEEE international conference on data mining (ICDM). pp. 598–607 (2019)

14. Zhang, G., Li, Z., Huang, J., Wu, J., Zhou, C., Yang, J., Gao, J.: efraudcom: An e-commerce fraud detection system via competitive graph neural networks. ACM Transactions on Information Systems (TOIS) **40**(3), 1–29 (2022)
15. Shi, C., Hu, B., Zhao, W.X., Philip, S.Y.: Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering **31**(2), 357–370 (2018)
16. Lu, Y., Shi, C., Hu, L., Liu, Z.: Relation structure-aware heterogeneous information network embedding. In: Proceedings of the AAAI conference on artificial intelligence. pp. 4456–4463 (2019)
17. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. Proceedings of the VLDB Endowment **4**(11), 992–1003 (2011)
18. Fu, T.y., Lee, W.C., Lei, Z.: Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 1797–1806 (2017)
19. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 315–323. JMLR Workshop and Conference Proceedings (2011)
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 6000–6010 (2017)
21. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 135–144 (2017)
22. Kingma, D.P.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X.: Applied logistic regression. John Wiley & Sons, Hoboken, NJ (2013)
24. Erdogan, B.E.: Prediction of bankruptcy using support vector machines: an application to bank bankruptcy. Journal of Statistical Computation and Simulation **83**(8), 1543–1555 (2013)
25. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of Statistics pp. 1189–1232 (2001)
26. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
27. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: The 6th International Conference on Learning Representations (ICLR) (2018)
28. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI conference on artificial intelligence. pp. 922–929 (2019)
29. Sun, X., Yin, H., Liu, B., Chen, H., Cao, J., Shao, Y., Viet Hung, N.Q.: Heterogeneous hypergraph embedding for graph classification. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM). pp. 725–733 (2021)
30. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: The 15th International Conference on Extended Semantic Web Conference. pp. 593–607 (2018)
31. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: International World Wide Web Conference (WWW). pp. 2022–2032 (2019)