Heterogeneous Graph Attention Network for Small and Medium-Sized Enterprises Bankruptcy Prediction

Yizhen Zheng[1(✉)], Vincent C. S. Lee[1], Zonghan Wu[2], and Shirui Pan[1]

[1] Department of Data Science and AI, Faculty of IT, Monash University, Melbourne, Australia
yzhe0006@student.monash.edu, {vincent.cs.lee,shirui.pan}@monash.edu
[2] University of Technology Sydney, Ultimo, Australia
zonghan.wu-3@student.uts.edu.au

Abstract. Credit assessment for Small and Medium-sized Enterprises (SMEs) is of great interest to financial institutions such as commercial banks and Peer-to-Peer lending platforms. Effective credit rating modeling can help them make loan-granted decisions while limiting their risk exposure. Despite a substantial amount of research being conducted in this domain, there are three existing issues. Firstly, many of them are mainly developed based on financial statements, which usually are not publicly-accessible for SMEs. Secondly, they always neglect the rich relational information embodied in financial networks. Finally, existing graph-neural-network-based (GNN) approaches for credit assessment are only applicable to homogeneous networks. To address these issues, we propose a heterogeneous-attention-network-based model (HAT) to facilitate SMEs bankruptcy prediction using publicly-accessible data. Specifically, our model has two major components: a heterogeneous neighborhood encoding layer and a triple attention output layer. While the first layer can encapsulate target nodes' heterogeneous neighborhood information to address the graph heterogeneity, the latter can generate the prediction by considering the importance of different metapath-based neighbors, metapaths, and networks. Extensive experiments in a real-world dataset demonstrate the effectiveness of our model compared with baselines.

Keywords: Bankruptcy prediction · Financial network · Heterogeneous graph · Graph neural networks · Graph attention networks

1 Introduction

Representing about 90% of the business entities and 50% of employment worldwide, small and medium-sized enterprises (SMEs) play an essential role in the economy [1]. However, many of them face huge financing obstacles, especially when applying for loans from commercial banks, due to their financial status's

K. Karlapalem et al. (Eds.): PAKDD 2021, LNAI 12712, pp. 140–151, 2021.
https://doi.org/10.1007/978-3-030-75762-5_12

opacity. One of the reasons is that SMEs are usually not listed, and thus, they do not have the accountability to publish their financial reports and statements regularly. Therefore, there is a dearth of public SMEs' financial data, and it poses a challenge on credit risk assessment tasks for them. The other reason is that even though financial professionals in banks can get access to SMEs' financial statements, in practice, SMEs may intentionally beautify their statements and commit accounting fraud to meet the issuer's criteria. As a result, credit issuers bear the significant risk of uncertainty in lending to SMEs and thus are reluctant to accept their financing applications.

Most existing credit risk assessing methods are based on traditional machine learning algorithms using structured financial statement data [5,11]. However, it is difficult to obtain SMEs' financial statements, which poses a significant challenge for SMEs' credit risk assessment. Also, these traditional machine learning approaches cannot learn the rich relational information embodied in financial graphs. Financial graphs provide rich relational information, which is valuable in inferring a firm's credit condition since its financial status can be affected by its connections, such as its shareholders and executives. In recent years, some studies are attempting to apply GNN-based models to exploiting financial networks. However, these methods are developed on homogeneous financial networks such as company-to-company guarantee networks [4,13]. This kind of network has only one type of node. There are no GNN-based approaches tailored for heterogeneous financial networks which contain multiple node types and edge types. Though there are some general heterogeneous information network (HIN) embedding methods, such as HAN, HetGNN, and MAGNN [6,17,19], which can learn node embeddings on heterogeneous graphs, these methods cannot be directly applied for credit risk assessment tasks. First, these models can only handle node information from a single source or graph, whereas for credit risk assessing a node's information may need to be retrieved and processed from multiple heterogeneous financial graphs. Second, these methods cannot effectively capture the information in target nodes' heterogeneous neighborhoods. This is because they only use a transformation matrix to project the features of different types of nodes into the same vector space to address nodes' heterogeneity instead of sufficiently learning the heterogeneous neighborhood for each node.

We propose a new heterogeneous graph attention network-based approach (HAT) for SMEs' practical risk assessment by using publicly accessible data to address the issues mentioned above. This model can adequately utilize the features of different types of nodes and relations from multiple heterogeneous data sources to predict companies' bankruptcy. Specifically, we first use a random walk with a restart mechanism (RWR) to find a fixed-size of strongly correlated type-specific neighbors for each heterogeneous neighbor type for target nodes. Then, we use the attention mechanism to aggregate heterogeneous type-specific neighbor's features and concatenate the output with target nodes' features to obtain target nodes' heterogeneous neighborhood-level embeddings. In this way, information from all heterogeneous neighbors can be embedded with target nodes' representation. Following, with target nodes' heterogeneous neighborhood-level

embeddings, we use the triple attention output layer to train an end-to-end SMEs' bankruptcy predictor. Using triple-level attention, our model can consider the importance of different metapath-based neighbors, metapaths, and graph data sources. To evaluate our proposed model's performance, we build a real-world dataset by collecting SMEs' public data from multiple open data sources. Then, we use this data to construct two heterogeneous financial graphs: a shareholder network and a board member and executive network. The effectiveness of our proposed model has been verified through extensive experiments. In a nutshell, the main contributions of this paper can be summarized as follows:

1. To the best of our knowledge, this is the first attempt to approach bankruptcy prediction using heterogeneous graph neural network techniques. Our work opens a way to develop the company's credit risk evaluation solutions considering rich information embodied in HINs.
2. We propose HAT, a heterogeneous attention network, which can effectively encode information of multiple types of node features and relations from multiple graph data sources to evaluate SMEs' credit risk.
3. We have conducted extensive experiments on a self-collected real-world dataset. The result demonstrates and verifies the effectiveness of our method.

2 Related Work

Bankruptcy Prediction. Recently, researchers have developed machine learning-based bankruptcy prediction models using financial ratio data or privately-owned data provided by financial institutions. For example, Mai et al. [11] collect accounting data for 11827 US-listed firms to train a deep learning model with an average embedding layer to forecast bankruptcy. Chen et al. [3] exploited the ensemble proportion learning model with SVM based on four public bankruptcy datasets on the UCI repository. However, these methods relied heavily on financial ratios, which can be distorted with accounting beautifying techniques such as "window dressing". Also, this data is usually not publicly accessible for SMEs. Though some studies conducted their research on public bankruptcy datasets, their number is very limited and always out-of-date. Furthermore, these studies have not exploited financial graphs, whose rich relational information can provide valuable clues when inferring firms' credit status.

Graph Neural Networks. Exploiting graph-structured data with deep learning, graph neural networks (GNN), can effectively generate a low-dimensional vector representation for each network node [16,18,20]. These latent representations can then be used for various downstream tasks, such as node classification and link prediction [9]. Extending the graph convolutional networks (GCNs) with masked self-attention layers, Velickovic et al. [15] proposed Graph attention networks (GAT) to consider neighbor nodes' weights, rather than treating neighbors' nodes equally important. In recent years, some pioneering studies are implementing GNN-based approaches in financial networks. For instance,

Cheng et al. [4] proposed a high-order graph attention representation method to infer the systematic credit risk based on company-to-company guarantee networks. Shumovskaia et al. [13] developed a recursive-neural-network-based GNN model to explore bank-client transactional networks. However, there is no study applying the GNN-based approach in financial HINs. This paper introduces a novel model, HAT, to learn node embeddings in financial HINs effectively.

3 Problem Formulation

Financial Heterogeneous Graph. This paper considers two financial networks, including shareholder networks, representing firms' shareholding structure and board member networks, showing firms' board members and executives as heterogeneous graphs. Figure 1 and Fig. 2 show examples for these two financial graphs. For a heterogeneous graph $G = (V, E)$, V and E represent nodes and edges in the graph, respectively. Here, node V includes two types of nodes: companies C and individuals I. The edges consist of company-individual edges E_{ci} and company-company edges E_{cc}.

Problem Statement. Given a set of heterogeneous financial graphs $\{g_1, g_2..., g_n\}$, in which each graph is defined as $G = (V, E)$, we aim to predict whether a firm will go bankrupt by learning a global node embedding $\mathbf{h}_c$ for each firm node c and mapping it to a probability distribution.

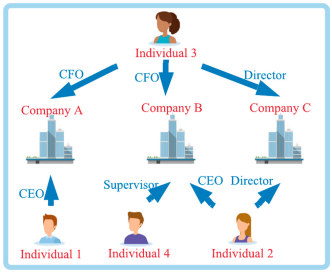

Fig. 1. An example of board member and executives' networks

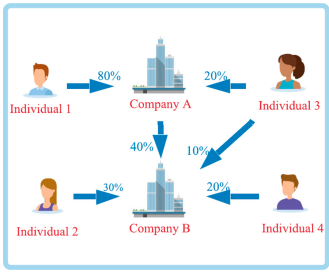

Fig. 2. An example of shareholder networks

4 Methodology of Our Proposed Model

In this section, we describe HAT for heterogeneous graph embedding learning in financial networks. As shown in Fig. 3, HAT consists of two main components: a heterogeneous neighborhood encoding layer and a triple attention output layer. While the heterogeneous neighborhood encoding layer can sufficiently exploit the heterogeneous neighborhood for target nodes, the triple attention and output layer can generate the final embedding and prediction results with three levels of attention: node-level, metapath-level, and network-level.

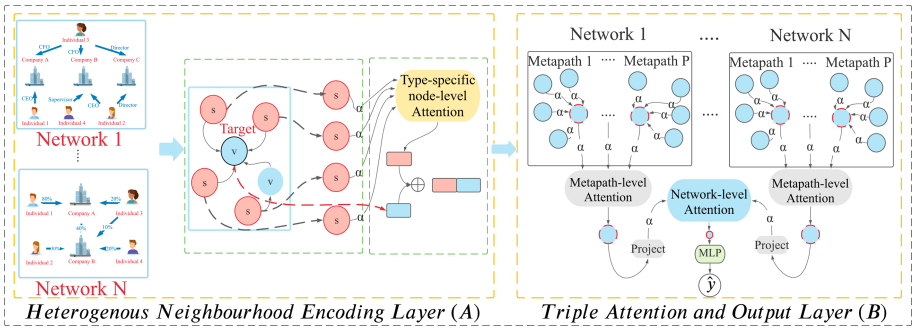

Fig. 3. The architecture of HAT for bankruptcy prediction

4.1 Heterogeneous Neighborhood Encoding Layer

A heterogeneous neighborhood encoding layer can address the heterogeneity of a heterogeneous graph. This layer first uses the RWR method to find each heterogeneous type top K type-specific neighbors for target nodes and aggregates them to get type-specific neighborhood embeddings. Then, it concatenates these type-specific neighborhood embeddings with target nodes' own features to obtain the all-type neighborhood embeddings.

It is challenging to solve heterogeneous graphs' heterogeneity since different types of nodes can have their features lie in different feature spaces. To solve this issue, there are two existing solutions. One is to concatenate features of different types of nodes to build a new large feature space, in which irrelevant dimensions for other types of nodes are assigned 0. The other solution uses a linear transformation matrix to project each kind of node's feature vectors into the same feature space [8,10]. However, these solutions may suffer from reduced performance because they include more noisy information or cannot sufficiently encode a node's heterogeneous neighborhood. Our solution first applies RWR to explore a fixed-size K of strongly correlated type-specific neighbors for target nodes. Here K is a tunable parameter. As shown in Fig. 3(A), given a node v and its $s \in S$ type neighbors, the two steps of this process are presented below:

1. Step-1: Sampling a fixed-length sequence based on RWR for the node v: We start a random walk from a node v. According to a probability p, the walk decides whether it will move to one of the current node's neighbors or restart from node v iteratively until it has collected a sequence with a preset fixed number of nodes, defined as RWR (v).
2. Step-2: Finding top K neighbors for v: Based on s type neighbors occurrence frequency on RWR (v), we select top K s type neighbors for v.

We can then aggregate these top K collected nodes' features to encode v's s type neighborhoods. If a target node has multiple heterogeneous neighbor types, we can repeat the two-step process for each neighbor type to generate each type neighborhood representation. Since each type-specific neighbor may contribute

differently to a target node, it is necessary to consider their weight in the aggregation process. Therefore, we use an attention layer in heterogeneous neighbor nodes' aggregation. Given a fixed number K and a heterogeneous neighbor node type $s \in S$, for a target node, it has a set of type-specific neighbors features h_1^s, h_2^s, $\ldots h_K^s$. With these features, we can use an attention layer to learn the weight of each type-specific neighbor α_1^s, α_2^s, $\ldots \alpha_K^s$ and this process can be formulated as follows:

$$\{\alpha_1^s, \alpha_2^s, \ldots \alpha_K^s\} = att_{heterneigh}\{h_1^s, h_2^s, \ldots h_K^s\}. \tag{1}$$

In this equation, $att_{heterneigh}$ means the self-attention-based [14] deep neural network for type-based neighbors' aggregation. Specifically, we first use a one-layer MLP to transform these features into a heterogeneous type-specific node-level attention vector. Then, we can get the importance of each heterogeneous type-based neighbor, denoted as e_k^s, where $\{k|k \in Z, 0 \leq k \leq K\}$:

$$e_k^s = q_H^T \cdot tanh(W_H \cdot h_k^s + b_H), \tag{2}$$

where q_H is the parameterized attention vector, W_H is the weight matrix, and b_H is the bias vector. All of these parameters are learnable. Then, by using a softmax layer, we can normalize e_k^s to get the coefficient α_k^s:

$$\alpha_k^s = softmax(e_k^s) = \frac{exp(e_k^s)}{\sum_{k=1}^{K} exp(e_k^s)}, \tag{3}$$

here α_k^s represents the relative importance of a type-specific neighbor. With α_k^s, we can now polymerize the features of top K type-based neighbors for target nodes to obtain the learned embedding of node i's type s neighborhood. This process is presented as follows:

$$Z_i^s = \sum_{k=1}^{K} \alpha_k^s \cdot h_k^s, \tag{4}$$

where Z_i^s is the representation of node i's type s neighborhood. Finally, to obtain the embedding of node i's all type neighborhood, including node i's self-generated features, we need to aggregate the learning embedding of all Z_i^s with node i's own feature as follows:

$$Z_i^S = (\|_{s \in S} Z_i^s) \parallel h_i, \tag{5}$$

where $\parallel$ means concatenation, Z_i^S represents the all type neighborhood embedding for the node i, and h_i means the target node i's own feature. To better understand this aggregating and concatenation process, we present a graphical illustration in Fig. 4. From this figure, we can see node $c1$ has two types of heterogeneous neighbors: nn and ii. Through neighbor sampling, we can get the nn type and ii type top K neighbors for the node $c1$. Here $\{\alpha_1^{s1}, \alpha_2^{s1}, .., \alpha_4^{s1}\}$ and $\{\alpha_1^{s2}, \alpha_2^{s2}, .., \alpha_4^{s2}\}$ are two sets of computed attention coefficient for $c1$'s ii type neighbors and nn type neighbors respectively. After aggregation, node $c1$ gets its ii type neighborhood representation Z_{c1}^{s1}, and nn Z_{c1}^{s2}. Finally, by aggregating these two representations with c1's own feature h_{c1}. Node $c1$ get its all type neighborhood embedding Z_{c1}^S.

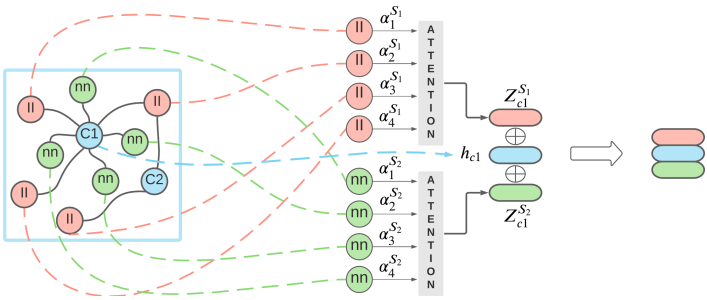

Fig. 4. Graphical illustration for the heterogeneous neighborhood encoding layer

4.2 Triple Attention and Output Layer

With the all-type neighborhood embeddings, the triple attention and output layer can generate the final embeddings for target nodes using triple-level attention: metapath-based node level, metapath level, and network level. The architecture of the triple-level attention layer is presented in Fig. 3(B).

Metapath-Based Node-Level Attention. For the metapath-based node-level attention, we adopt the multi-head graph attention operation [15], which is an attention mechanism for homogeneous graph representation learning. This operation is applicable here since, within a metapath, a target node is within a homogeneous graph with its same type neighbors. By using this operation, the model can learn the weight of different metapath-based neighbors.

A heterogeneous network has several metapaths denoted as $m \in M$ based on a target node type. Given a specific node i with type s, the node-level attention learns the weights of its metapath-based neighbor $j \in N_i^m$, where N_i^m represents a set of nodes including both node i's metapath-based neighbors and node i itself. Then, we compute the node-level attention value α_{ij}^m by using the all-type neighborhood embeddings Z_i^S and Z_j^S. With the normalized importance value α_{ij}^m, we can weight sum all the nodes in the set N_i^m to get the metapath-level embedding for node i. This process is formulated as follows:

$$\alpha_{ij}^m = \frac{exp(\sigma(q_m^T \cdot [Z_i^S \parallel Z_j^S]))}{\sum_{o \in N_i^m} exp(\sigma(q_m^T \cdot [Z_i^S \parallel Z_o^S]))}, \tag{6}$$

$$Z_i^m = \sigma(\sum_{j \in N_i^m} \alpha_{ij}^m \cdot Z_j^S), \tag{7}$$

where Z_i^m denotes the m metapath-based embedding for node i. To stabilize this learning process, multi-head attention can be used to improve the representation ability. Specifically, we can repeat the node-level attention L times to get L embeddings for node i. Then, the metapath-specific embedding can be generated by concatenating these embeddings:

$$Z_i^m = \|_{l=1}^{L} \sigma(\sum_{j \in N_i^m} \alpha_{ij}^m \cdot Z_j^S). \tag{8}$$

After learning the embedding for each metapath m, we can get a set of Z_i^m for the node i. This set of embedding will be aggregated through the metapath-level attention to get the network-level embedding for i.

Metapath-Level Attention. We introduce a self-attention-based metapath-level attention [14] to compute the weights for different metapaths. Given a specific node i with type s, it has several metapath-specific embeddings Z_i^m for several meta paths $m \in M$. With this set of semantic-specific embeddings, we can calculate the normalized metapath level attention α_m^P. Please note that, here α_m^P is shared by all nodes, since there is some similar connection between a metapath and a type of nodes. Then, we fuse metapath-specific embeddings of i by weighted summing all Z_i^m with α_m^P, to create the network-level embedding:

$$\alpha_m^P = \frac{exp(\frac{1}{|V_s|}\sum_{i \in V_s} q_O^T tanh(W_O \cdot Z_i^m + b_O))}{\sum_{m \in M} exp(\frac{1}{|V_s|}\sum_{i \in V_s} q_O^T tanh(W_O \cdot Z_i^m + b_O))}, \tag{9}$$

$$Z_i^g = \sum_{m \in M} \alpha_m^P \cdot Z_i^m, \tag{10}$$

here Z_i^g is the network-specific embedding for the node i, q_O is a parameterized vector, W_O is a weight matrix, and b_O is a bias vector. With all Z_i^g, we can polymerize them to get node i's final embedding via network-level attention.

Network-Level Attention. Network-level attention can address the heterogeneity of different networks and determine their importance. Firstly, we project network-level embeddings to the same vector space with a linear transformation layer. Given a node i with type s, and $g \in G$, where G is the set of graphs input, the process of projection is shown below:

$$Z_i^{g'} = \theta_s \cdot Z_i^g, \tag{11}$$

where $Z_i^{g'}$ is the projected network-level embeddings, Z_i^g is the original network-level embeddings, and θ_s is the transformation matrix for the node type s. Like metapath-level attention, network-level attention applies the attention mechanism to distill the network semantic information from multiple graphs. Given a node i with type s, and a network $g \in G$, we can use the transformed embedding $Z_i^{g'}$ to compute the normalized network-level attention α_g. Then, by weighted sum all $Z_i^{g'}$, we can get the final embedding for node i. This process is formulated as below:

$$\alpha_n = \frac{exp(\frac{1}{|V_s|}\sum_{i \in V_s} q_R^T \cdot tanh(W_R \cdot Z_i^{g'} + b_R))}{\sum_{g \in G} exp(\frac{1}{|V_s|}\sum_{i \in V_s} q_R^T \cdot tanh(W_R \cdot Z_i^{g'} + b_R))}, \tag{12}$$

$$Z_i = \sum_{g \in G} \alpha_g \cdot Z_i^{g'}, \tag{13}$$

here Z_i is the final embedding for node i, q_R is a parameterized vector, W_R is a weight matrix, and b_R is a bias vector. Z_i can be transformed to the desired output dimension for further tasks.

4.3 Training

We consider bankruptcy prediction as a binary classification task. The output layer predicts the labels of companies based on its final embedding Z_i. To obtain the output, we feed the embedding to a softmax layer for classification as follows:

$$Z_i = softmax(Z_i) \tag{14}$$

Then, we can optimize the model parameters by minimizing the cross-entropy loss over training data with the L2-norm:

$$L = -\sum_{i \in V_s} Y_i log(C \cdot Z_i) + \eta \parallel \theta \parallel_2 \tag{15}$$

where C is the parameter of the classifier, $i \in V_s$ represents a node with the type s, Y_i is the ground truth label for node i, Z_i is i's node embedding, θ is the model parameters, and η represents the regularization factor. With node labels' guidance, we adopt the gradient descent method to process the backpropagation and learn the node embeddings for target nodes.

5 Experiment

Datasets. We collect data from multiple Chinese government open data sources and build a real-world dataset containing a board member network and a shareholder network for 13489 companies. In the dataset, 3566 companies are labeled as bankrupt. The experiment source code and collected dataset are available in our Github repository[1].

Experiment Methods. We select five baselines to verify the effectiveness of our proposed method.

- **Logistic Regression (LR).** LR is a statistical model powered by a shallow neural network. It is widely adopted in bankruptcy prediction tasks [7].
- **Support Vector Machine (SVM).** SVM is a machine learning technique, which builds hyperplanes in a high dimensional space for classification and regression tasks. It is widely adopted in bankruptcy prediction tasks [2].

[1] https://github.com/hetergraphforbankruptcypredict/HAT.

Table 1. Experiment result

Method	Accuracy	Macro-F1	Recall	Precision
LR	76.03	62.83	61.63	68.51
SVM	75.89	62.45	61.31	68.25
DeepWalk	60.30	50.97	51.22	51.09
GAT	76.50	62.10	60.98	69.95
HAN	77.06	63.07	61.75	71.22
HAT	**78.31**	**64.32**	**62.72**	**74.86**

Table 2. Variants result

Method	Accuracy	Macro-F1	Recall	Precision
HAT_{mean}	78.08	63.41	61.99	74.80
HAT_{max}	77.61	64.42	62.82	72.23
HAT_{min}	78.22	64.51	62.89	74.30
HAT_{dual}	78.26	**66.18**	**64.40**	73.18
HAT	**78.31**	64.32	62.72	**74.86**

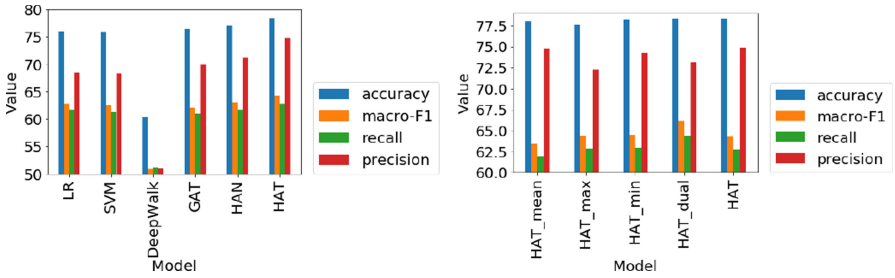

Fig. 5. Comparison of baselines and variants of HAT.

- **DeepWalk (DW).** DW is a random-walk-based graph embedding method for homogeneous graphs. In the experiment, we ignore the heterogeneous graph's heterogeneity and process the DW through the whole graph [12].
- **GAT.** GAT is a semi-supervised attention-based GNN approach for homogeneous graphs. Here we only consider the best metapath [15].
- **HAN.** HAN is a state-of-the-art heterogeneous graph neural network with dual-level attention. It is a general heterogeneous GNN method [17].

Evaluation Metrics. We have selected four metrics: Micro-F1, Macro-F1, Recall, and Precision to evaluate the performance of our model.

Implementation Details. In the experiment, we consider the data split based on the bankruptcy date of company nodes. While the training set includes all nodes that went bankrupt before 2019, the validation and test set include nodes that went bankrupt after 2019. This is reasonable because the model's goal is to predict companies' bankruptcy. For active company nodes, we distribute them according to the ratio 68:16:16 for training, validation, and test set, respectively. This is because the training set contains 68% of all bankruptcy nodes in our dataset, while the remaining bankruptcy nodes are distributed equally for the other two sets. Thus, with this ratio, the data distribution in all three sets would be similar.

Results and Analysis. The performance of HAT and baseline methods is presented in Table 1 and Fig. 5. From this table, we have the following observations:

- Surprisingly, GNN-based method GAT only obtain similar results to LR and SVM, the reason may be that GAT only uses one metapath-based homogeneous graph as input, which cannot provide sufficient graph structural and relational information to advance the model training.
- HAN outperforms the other four baselines, which is reasonable because HAN can learn additional useful signals provided in HINs.
- HAT achieves the best performance in all evaluation metrics. This result shows the superiority of our approach to traditional machine learning and state-of-the-art GNN-based approaches in exploring heterogeneous financial networks. This improvement could be attributed to heterogeneous neighborhood learning and the triple-level attention in neighbor aggregation.

Ablation Study. We have conducted experiments for four HAT variants as comparisons, and the results are shown in Table 2.

- **HAT**$_{mean}$**:** Compared with HAN, HAT$_{mean}$ has the heterogeneous neighborhood encoding layer and use mean pooling in neighbors aggregation. Also, it applies the triple-level attention instead of the dual-level attention.
- **HAT**$_{max}$**:** Compared with HAT$_{mean}$, HAT$_{max}$ applys max pooling in the heterogeneous neighborhood encoding layer.
- **HAT**$_{min}$**:** Compared with HAT$_{mean}$, HAT$_{min}$ adopts min pooling in the heterogeneous neighborhood encoding layer.
- **HAT**$_{dual}$**:** Compared with HAT, HAT$_{dual}$ has no network-level attention.
- **HAT:** HAT employs an attention mechanism in the heterogeneous neighborhood encoding layer.

From Table 2 and Fig. 5, we can see that all HAT variants outperform HAN in all evaluation metrics. This result demonstrates the heterogeneous neighborhood encoding layer's effectiveness, which can encapsulate the heterogeneous neighborhood information to boost the model performance. Also, while HAT$_{dual}$ achieves the best performance for accuracy and precision, HAT obtains the highest results for Macro-F1 and Recall. It is hard to tell which one is better between HAT$_{dual}$ and HAT in the scenario in which heterogeneous graph inputs have the same set of node types. However, suppose graph inputs have a different set of node types. In that case, only HAT can address the heterogeneity of different networks since it has additional network-level attention to project network-level embeddings. Therefore, HAT is preferable due to its comprehensiveness in handling graph inputs.

6 Conclusion

This paper develops a novel heterogeneous GNN-based method HAT for SMEs' bankruptcy prediction using public data from multiple sources. Our model can

effectively learn the heterogeneous neighborhood for target nodes and generate node embeddings considering the weights of metapath-based neighbors, metapaths, and networks, with a triple attention output layer. The results of extensive experiments have shown that our model outperforms five baselines in exploring financial HINs. For future works, we plan to apply our method to other scenarios, e.g., social and e-commerce networks.

References

1. Small and Medium Enterprises (SME) finance (2020). https://www.worldbank.org/en/topic/smefinance. Accessed 4 Nov 2020
2. Chaudhuri, A., De, K.: Fuzzy support vector machine for bankruptcy prediction. Appl. Soft Comput. **11**(2), 2472–2486 (2011)
3. Chen, Z., Chen, W., Shi, Y.: Ensemble learning with label proportions for bankruptcy prediction. Expert Syst. Appl. **146**, 113115 (2020)
4. Cheng, D., Zhang, Y., Yang, F., Tu, Y., Niu, Z., Zhang, L.: A dynamic default prediction framework for networked-guarantee loans. In: CIKM (2019)
5. Erdogan, B.E.: Prediction of bankruptcy using support vector machines: an application to bank bankruptcy. J. Stat. Comput. Simul. **83**(8), 1543–1555 (2013)
6. Fu, X., Zhang, J., Meng, Z., King, I.: MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. In: WWW 2020 (2020)
7. Hauser, R.P., Booth, D.: Predicting bankruptcy with robust logistic regression. J. Data Sci. **9**(4), 565–584 (2011)
8. Huang, Q., Yu, J., Wu, J., Wang, B.: Heterogeneous graph attention networks for early detection of rumors on Twitter. In: IJCNN (2020)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2016)
10. Linmei, H., Yang, T., Shi, C., Ji, H., Li, X.: Heterogeneous graph attention networks for semi-supervised short text classification. In: EMNLP-IJCNLP (2019)
11. Mai, F., Tian, S., Lee, C., Ma, L.: Deep learning models for bankruptcy prediction using textual disclosures. Eur. J. Oper. Res. **274**(2), 743–758 (2019)
12. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: KDD (2014)
13. Shumovskaia, V., Fedyanin, K., Sukharev, I., Berestnev, D., Panov, M.: Linking bank clients using graph neural networks powered by rich transactional data (2020)
14. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
15. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: ICLR (2017)
16. Wang, H., Zhou, C., Chen, X., Wu, J., Pan, S., Wang, J.: Graph stochastic neural networks for semi-supervised learning. In: NeurIPS (2020)
17. Wang, X., et al.: Heterogeneous graph attention network. In: WWW (2019)
18. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. In: IEEE Transactions on Neural Networks and Learning Systems (2020)
19. Zhang, C., Song, D., Huang, C., Swami, A., Chawla, N.V.: Heterogeneous graph neural network. In: SIGKDD (2019)
20. Zhu, S., Pan, S., Zhou, C., Wu, J., Cao, Y., Wang, B.: Graph geometry interaction learning. In: NeurIPS (2020)