

# Risk Assessment for Networked-guarantee Loans Using High-order Graph Attention Representation

Dawei Cheng<sup>1</sup>, Yi Tu<sup>1</sup>, Zhenwei Ma<sup>2</sup>, Zhibin Niu<sup>3</sup> and Liqing Zhang<sup>1\*</sup>

<sup>1</sup>Department of Computer Science and Engineering,  
Shanghai Jiao Tong University, Shanghai, China

<sup>2</sup>School of Mathematical Science, Shanghai Jiao Tong University, Shanghai, China

<sup>3</sup>College of Intelligence and Computing, Tianjin University, Tianjin, China

{dawei.cheng, tuyi1991, zhenweima}@sjtu.edu.cn, zniu@tju.edu.cn, zhang-lq@cs.sjtu.edu.cn

## Abstract

Assessing and predicting the default risk of networked-guarantee loans is critical for the commercial banks and financial regulatory authorities. The guarantee relationships between the loan companies are usually modeled as directed networks. Learning the informative low-dimensional representation of the networks is important for the default risk prediction of loan companies, even for the assessment of systematic financial risk level. In this paper, we propose a high-order graph attention representation method (HGAR) to learn the embedding of guarantee networks. Because this financial network is different from other complex networks, such as social, language, or citation networks, we set the binary roles of vertices and define high-order adjacent measures based on financial domain characteristics. We design objective functions in addition to a graph attention layer to capture the importance of nodes. We implement a productive learning strategy and prove that the complexity is near-linear with the number of edges, which could scale to large datasets. Extensive experiments demonstrate the superiority of our model over state-of-the-art method. We also evaluate the model in a real-world loan risk control system, and the results validate the effectiveness of our proposed approaches.

## 1 Introduction

The bank loans are critical for the development of the small and medium enterprise [Biggs, 2002]. As the traditional rules are usually designed for the majority independent market players, the existing loan assessment criteria is inadequate for real demand. Companies are allowed to guarantee each other to enhance loan security. When more and more companies are involved, they form complex guarantee networks. Such complex structures are a double-edged sword for the national economy. On one hand, such easy-loans can help companies raise funds quickly, on the other hand, highly coupled relationships could magnify default risk along the guarantee

chain even lead to massive defaults systematic risk. Whether a loan will be delinquent depends not only on its own financial status, but also on other companies in the network, which brings new challenges in loan risk assessment for banks.

Traditional approaches to predicting bank loan delinquency mainly use financial probabilistic models [Siddiqi, 2012; Bravo *et al.*, 2015], for example the credit scorecards, which utilize a shallow linear regression or classification model with the borrower's financial information. With recent advances in machine learning techniques, different algorithms have emerged to improve prediction accuracy by increasing model capacity and adding behavior information [Fitzpatrick and Mues, 2016; Cheng *et al.*, 2018].

However, these techniques (credit scorecards and machine learning methods) have two significant disadvantages: 1) they rely mainly on historical information of individual enterprises and are therefore unable to discover the delinquency probability caused by networked loans; 2) the classification model gives equal weights to each sample (i.e., each SME), but different nodes contribute differently in networked-guarantee loans. That means we should focus on dominant companies to curb systemic risk so as to discover the risk patterns.

To this end, we propose a high-order graph attention representation method (HGAR) for the default risk assessment of the networked-guarantee loans. The intuition behind the technique is that, based on our observations, the default probability of a borrower involved in a guarantee network is also influenced by connected nodes. This is because, in guarantee loans, if the borrower fails to repay to the bank, its guarantors have the debt obligation. This means defaults may spread rapidly across the network like the infectious disease transmission, especially during the period of economic downturn where many companies will fail to repay loans. This would have a domino effect and cause numerous enterprises to fall into an unfavorable situation.

Therefore, by integrating guarantee network representations to loan risk assessment systems, we are able to infer the delinquent probability comprehensively. We first set each vertex in the guarantee network to a binary role, guarantor or guarantee, and then present a graph attention layer that captures nodes' dominant importance in a network. Afterwards, we propose objective functions to preserve the binary role of nodes and high-order adjacency, and apply a productive strat-

\*Corresponding Author

egy to optimize it. Through theoretical analysis, we demonstrate that the complexity of the proposed method is linear to the number of edges, which could scale to real-world large networks.

In a nutshell, the main contributions of this paper are:

1. We provide the first attempt in loan risk assessment by adding high-order graph attention representations (HGAR) in order to predict loan delinquency and warn of a domino effect loan crisis.
2. We propose a new high-order and graph attention based network representation method and demonstrate that the learned representation can improve the performance of loan delinquency prediction.
3. We conduct extensive experiments on real-world datasets over six months’ empirical study. The result demonstrates that our methods can effectively predict defaults and provide early warning of loan risks.

## 2 Related Works

We summarize the related works in two main areas: 1) graph representation; 2) networked financial risk.

### Graph Representation

Graph representation is also known as network embedding, which is a learning approach designed to map each node of a network to a low-dimensional space by machine learning techniques [Perozzi *et al.*, 2014; Cheng *et al.*, 2019]. It learns latent representations using local information obtained from various types of walks, which are treated as the equivalent of sentences [Velickovic *et al.*, 2018]; alternatively, it represents the connections between network vertices in the form of a matrix and uses different matrix factorization methods to obtain the representations [Liao *et al.*, 2018; Li *et al.*, 2018]. However, most of these studies are targeted on social, semantic, or citation networks, dismissing domain characteristics. Few of them tackle the loan defaults problem in financial networks.

### Networked Financial Risk

Financial crises and systemic risk have always been a major concern for financial companies and governments, with extensive work on this having been undertaken [Fischer and Molenaar, 2012]. Networks represented by interconnected nodes and links between them are a good representation of modern financial systems as they also have complex interdependence and connections internally [Allen and Babus, 2008]. Researchers started to pay more attention on network theory after the 2008 global financial crisis, because the crisis brought on by the collapse of Lehman Brothers spread throughout connected corporations is similar to the epidemic of Severe Acute Respiratory Syndrome (SARS) in 2002. Both of them hit a networked system and caused cascaded consequences [Bougheas and Kirman, 2015]. Although initial efforts have been made using network theory to understand fundamental problems [Van Vlasselaer *et al.*, 2013], there is little work on risk assessment in guarantee loan networks except for the preliminary work [Meng *et al.*, 2017; Niu *et al.*, 2018].

## 3 Loan Risk Assessment Methods

### 3.1 Problem Definition

We first present the preliminary concepts and definitions and then introduce the main problem in this paper.

**Definition 3.1: Guarantee Network.** A guarantee network (GN) is a directed graph  $G = (V, E)$ , where  $V = \{v_1, v_2, \dots, v_m\}$  is a set of vertices and each vertex denotes an SME, and  $E = \{e_1, e_2, \dots, e_n\}$  is a set of guarantee relationships (edges).

**Definition 3.2: Loan Default(Delinquency) Event.** A loan default event in this paper refers to an SME’s delinquency of overdue repayments.

We now formalize our risk prediction problem using high-order graph attention representations as follows:

*Given a set of loan and repayment records, guarantee networks  $G = (V, E)$ , for each SME, this paper aims to present them in a low-dimensional space  $\mathbb{R}^d$ . We want to infer the possibility of loan default events based on learned low-dimensional representations and loan features. The objective is to achieve a high accuracy of default prediction as well as to explore the risk patterns of guaranteed loans.*

### 3.2 Graph Attentional Layer

Following Velickovic’s work [Velickovic *et al.*, 2018], we introduce graph attention layer in HGAR – an attention layer in neural networks that learns higher-level representations. Specifically, for a given network, we denote the latent embedding of node  $v_i$  as  $\vec{u}_i$  and the attentional embeddings as  $\vec{w}_i$ .

The input of the graph attention layer is the latent embeddings of nodes,  $u = \{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_N\}$ ,  $\vec{u}_i \in \mathbb{R}^F$ , where  $N$  is the number of nodes and  $F$  is the dimension of latent embeddings. The layer outputs attentional features,  $u' = \{\vec{u}'_1, \vec{u}'_2, \dots, \vec{u}'_N\}$ ,  $\vec{u}'_i \in \mathbb{R}^{F'}$ .

Particularly, we apply a shared linear transformation in an initial step in order to transform the input features into high-level representation. Each node is multiplied by a weighted matrix,  $W \in \mathbb{R}^{F' \times F}$ , and then a shared attentional mechanism  $\phi : (\mathbb{R}^{F'}, \mathbb{R}^{F'}) \rightarrow \mathbb{R}$  is applied on top of it. The attention coefficients indicate the importance of node  $v_j$ ’s feature to node  $v_i$ , denoted as:

$$e_{ij} = \phi(W\vec{u}_i, W\vec{u}_j) \tag{1}$$

Normally, the model involves all vertices in the graph, which implies for each node, we need to update its attention weights on every other node. In this paper, we inject the graph structure by only computing  $e_{i,j}$  for nodes  $v_j \in \mathcal{N}_i$ , where  $\mathcal{N}_i$  are the  $n$ -th order neighborhoods. We then normalize  $e_{i,j}$  across all choices of  $v_j$  by using the softmax function:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})} \tag{2}$$

By obtaining the normalized attention coefficients, which are parameters of a single-layer feed-forward neural network, we utilize them to compute the linear combination of the latent embeddings. Therefore, we reach the output feature as:

$$\vec{u}'_i = \sigma \left( \sum_{j \in \mathcal{N}_i} \alpha_{ij} W \vec{u}_j \right) \quad (3)$$

In order to stabilize the learning process of self-attention, we employ multi-head attention to the final layer of the network.  $K$  independent attention mechanisms execute as Equation 3. Their features are then aggregated, resulting in the following output feature representation:

$$\vec{u}'_i = \sigma \left( \frac{1}{K} \sum_{k=1}^K \sum_{j \in \mathcal{N}_i} \alpha_{ij} W^k \vec{u}_j \right) \quad (4)$$

### 3.3 High-order Embedding Methods

For given nodes  $v_i$  and  $v_j$ , if  $v_i$  provides a guarantee to node  $v_j$ , we perform an edge from  $v_i$  to  $v_j$ , indicating first-order neighborhoods, and we denote its adjacency as  $A_{i,j}$ . Consequently, we define the distance between vertex  $v_i$  and vertex  $v_j$  as  $r_{i,j} = \sum_l A_{i,j}^l$ , where  $l$  is the adjacent order index.

For each pair of vertices  $v_i$  and  $v_j$ , which connects in the first-order, we define its joint probability distribution as:

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-\vec{u}'_i \cdot \vec{u}'_j)} \quad (5)$$

where  $\vec{u}'_i \in \mathbb{R}^d$  is the attentional vector representation of vertex  $v_i$  in low-dimensional space, and  $\hat{p}_1(\cdot, \cdot)$  is a distribution on space  $V \times V$ . Then, we define the empirical distribution of  $\hat{p}_1$  as:

$$\hat{p}_1(v_i, v_j) = \frac{r_{i,j}}{\sum_{(i,j) \in cus} r_{i,j}} \quad (6)$$

The low-dimensional vector representation of vertices is optimized by minimizing the distance between the above two distributions:  $O_1 = d(\hat{p}_1(\cdot, \cdot), p_1(\cdot, \cdot))$ , where  $d(\cdot, \cdot)$  denotes the distance between  $p_1$  and  $\hat{p}_1$ .

We apply the widely used KL-divergence as the measure of distance between two distributions [Kompass, 2007]. Then, we get the distance as below:

$$d_{KL}(\hat{p}_1 \| p_1) \sim - \sum_{(i,j) \in cus} r_{i,j} \log p_1(v_i, v_j) \quad (7)$$

After omitting some constants, we reach the object function as:

$$\min O_1 \sim - \sum_{(i,j) \in cus} r_{i,j} \log \frac{1}{1 + \exp(-\vec{u}'_i \cdot \vec{u}'_j)} \quad (8)$$

In high-order estimation, we assume that vertices which have the same connections or similar structure are closer in learned feature space. This means each vertex is considered as both a vertex and a ‘‘context’’ of other vertices in the network. Specifically, we denote  $\vec{u}'_i$  as the attentional representation of a vertex and introduce  $\vec{u}''_i$  as the learned features when vertex  $i$  is considered as the ‘‘context’’ of the other vertex. Thus, for each pair of vertices  $(v_i, v_j)$ , given  $v_i$ , the conditional probability is expressed as:

$$p(v_j | v_i) = \frac{\exp(-\vec{u}''_j \cdot \vec{u}'_i)}{\sum_{k=1}^{|V|} \exp(-\vec{u}''_k \cdot \vec{u}'_i)} \quad (9)$$

where  $|V|$  is the number of vertices which share neighbors with  $v_i$ .

Also, we preserved network structure features between vertex  $v_i$  and  $v_j$  in the objective function. particularly, 1) the numbers of its guarantors; 2) the number of shared neighborhoods. Then, the probability between  $v_i$  and  $v_j$  is denoted as:

$$\hat{p}(v_j | v_i) = \frac{r_{i,j} + \rho(i, j)}{\sum_{k \in cus} (r_{i,k} + \rho(i, k))} \quad (10)$$

where  $\sum_{k \in cus} r_{i,k}$  is the collection of nodes that share the same neighborhoods with vertex  $v_i$ .

In order to preserve network structure information in a learned representation, we bring  $\rho(i, j)$  in Eq. 10, which is designed to describe the similarity of neighborhood information between vertex  $v_i$  and  $v_j$ . It is defined as follows:

$$\rho(i, j) = \lambda \cdot \left( \left| \sum_{k \in gua} r_{i,k} - \sum_{k \in gua} r_{j,k} \right| \right) \quad (11)$$

where  $\lambda$  is the penalty parameter to adjust the importance of the network structure in the empirical probability distribution.

We then learn the low-dimensional representation of each vertex  $v_i$  by minimizing the distance between the empirical conditional probability  $\hat{p}(\cdot | v_i)$  and estimated conditional probability  $p(\cdot | v_i)$ :

$$\min O_2 = \sum_{i \in V} \omega_i d(\hat{p}(\cdot | v_i), p(\cdot | v_i)) \quad (12)$$

where  $\omega_i$  denotes the number of the vertex  $v_i$ 's neighbors in networks. Same as the first-order estimation, we employ the KL-divergence as distance measures in Eq. 12. After simplifying the formation, we reach the object function as:

$$\min O_2 \sim - \sum_{(i,j) \in cus} (r_{i,j} + \rho(i, j)) \log \frac{\exp(-\vec{u}''_j \cdot \vec{u}'_i)}{\sum_{k=1}^{|V|} \exp(-\vec{u}''_k \cdot \vec{u}'_i)} \quad (13)$$

During training process, we include both of the first-order and high-order attentional representation. Afterwards, we concatenate the two estimated vectors as the final network low-dimensional representation in the downstream default prediction task.

### 3.4 Loan Default Prediction Layer

This paper aims to predict loan default events. After obtaining the representation of guarantee networks, we apply this to the prediction layer and the loss function is defined as:

$$\mathcal{L}(\theta) = - \frac{1}{N} \sum_{i=1}^N [y_i \log(\text{predict}(u'_i, s_i : \theta)) + (1 - y_i) \log(1 - \text{predict}(u'_i, s_i : \theta))] \quad (14)$$

where  $u'_i$  denotes the representation of the  $i^{th}$  node (SME), which is the output of the attention layer,  $s_i$  denotes the loan features, and  $y_i$  denotes the label of the  $i^{th}$  node, which is set to 1 if the loan defaults and 0 otherwise.  $\text{predict}(u'_i, s_i)$  is a prediction function that maps  $u'_i$  and  $s_i$  to a real valued score, indicating the probability of whether the company will default in the current time window. We implement  $\text{predict}(\cdot, \cdot; \theta)$  with another shallow neural network (two-layer Relu and one-layer sigmoid).

We extract loan features  $s_i$  from the user’s loan dynamic behavior. This includes: 1) active loan behavior, e.g., default status, loan type, loan amount, historical loan times, historical loan amounts, etc. They are calculated in the current time window; 2) historical behavior describes the accumulated features of a user’s loan behavior, such as total loan amount, loan times and default times; 3) the user profile contains company type, registered capital, number of employees, enterprise scale, etc. Most banks require a company to supply this basic information when the enterprise makes a loan application, and we use the latest version of the user profile.

### 3.5 Complexity Analysis

The calculation of objective function in Eq. 13 involves all vertices in the network, which requires the summation over all vertices when calculating the conditional probability  $p(v_j|v_i)$ . This is computationally inefficient, especially for large-scale networks. To address this problem, we employ the negative sampling approach, which samples multiple negative edges according to some noisy distribution for each edge  $e_{i,j}$ .

Specifically, sampling on a sequence of edges requires  $O(|E|)$  time, where  $|E|$  denotes the number of edges. During implementation, we draw a sample of edges by using the alias table method [Li *et al.*, 2014], which takes only  $O(1)$  time. Afterwards, the negative sampling approach in the optimization process takes  $O(d(T+1))$  time, where  $T$  is the number of negative samples. Therefore, it takes  $O(dT)$  time in each step in total. As the number of steps is proportional to the number of edges and feature dimensions, we reach the overall time complexity of HGAR as  $O(dTF|E|)$ . This demonstrates the complexity of our proposed method is linear to the number of edges, which does not depend on the number of vertices.

Meanwhile, the edge sampling method also speeds up the computation of the stochastic gradient descent process. In our implementation, we apply the Adam optimizer to update parameters. The initial learning rate is set to 0.0001, and the batch size is set to 128 by default.

## 4 Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness and efficiency of our proposed methods. We first describe the experimental settings. Then, we give the experiment results of delinquent prediction compared with other baselines. After that, the effects of the attentional layer and high-order embeddings are tested respectively. Finally, we provide the parameter sensitivity and scalability evaluation of the methods.

	AUC(2014)	AUC(2015)	AUC(2016)
LoanFeature	0.58107	0.58045	0.59461
GF	0.67605	0.66515	0.68440
DeepWalk	0.72536	0.72369	0.72914
LINE	0.73383	0.73509	0.73512
node2vec	0.73094	0.73082	0.73931
MNDF	0.73229	0.73855	0.74532
AANE	0.76505	0.76865	0.77228
SNE	0.76914	0.77670	0.78122
GATs	0.77009	0.77731	0.78281
HGAR-noho	0.77040	0.77746	0.78270
HGAR-noatt	0.77707	0.78191	0.78553
HGAR-all	<b>0.81310*</b>	<b>0.80988*</b>	<b>0.81875*</b>

Table 1: Comparison of the default prediction results.

### 4.1 Experimental Settings

We evaluate our method on the data from a major commercial bank in Asia, between 01/01/2013 and 31/12/2016. This includes 112872 nodes (companies), with 124957 edges (guarantee relationships). We found that most of the loans are repaid monthly. Hence, we aggregate the behavior feature with a one-month time window and mark the delinquency loans as the target label in the month.

#### Compared Methods

We use eight state-of-the-art methods to highlight the effectiveness of HGAR. These are: Graph Factorization (GF) [Ahmed *et al.*, 2013], DeepWalk [Perozzi *et al.*, 2014], LINE [Tang *et al.*, 2015], node2vec [Grover and Leskovec, 2016], MNDF [Wang *et al.*, 2017], AANE [Huang *et al.*, 2017], SNE [Liao *et al.*, 2018] and GATs [Velickovic *et al.*, 2018]. We also construct sub-modules of our proposed model in comparison: HGAR-noatt/noho/nolf, in which attention layer/high-order/loan features are not used. For LoanFeature, we only utilize loan features in the prediction task.

We set the network embedding dimensions to 64, combining loan features (64) with 128 dimensional vectors in the experiments. The parameters of the methods compared are set to their default values. For our method, we set  $\lambda = 1$  and  $l = 3$  since we often reach stable results with these. We apply the prediction layer for all methods and evaluate the performance by AUC (Area Under the Curve) and Precision@k. Precision@k means the prediction precision of the top k nodes, formulated as:

$$Precision@k = \frac{|\{i|i \in \mathbf{V}_p \cap \mathbf{V}_o\}|}{|\mathbf{V}_p|}$$

where  $\mathbf{V}_p$  is the set of predicted top  $k$  defaulted nodes,  $\mathbf{V}_o$  is the set of observed default nodes and  $|\cdot|$  represents the size of the set.

### 4.2 Loan Default Prediction

We evaluate the performance of different models for the default prediction task. Records from the year 2012 are used as training data, and then we predict defaults of the next three years in months. We report the average AUC in Table 1.

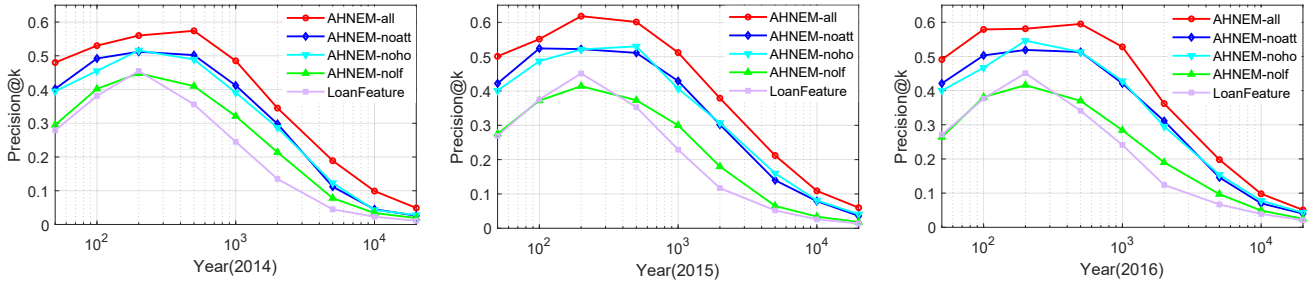


Figure 1: Precision@k results of default prediction. We compare HGAR-all with its sub-modules and LoanFeature. The result shows our proposed attentional high order embedding method achieves the best performance in all time windows.

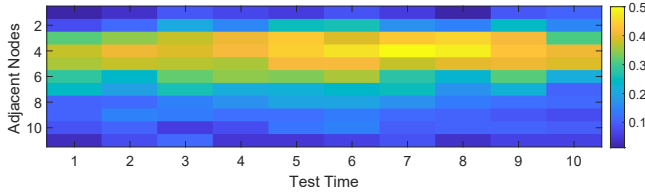


Figure 2: Case study results of the attention layer. The node with around 3-6 adjacent nodes is generally highlighted.

The first 9 rows of Table 1 report the results only of loan features and some state-of-the-art baselines. As we can see, eight baselines, which add network embeddings, outperform LoanFeature, showing improvements of at least 15%, demonstrating the effectiveness of adding guarantee network topology information in loan delinquency prediction. Of all the baselines, GATs and SNE are shown to be the most competitive. Rows 10–12 show the results of HGAR and some of its sub-models. The performance of HGAR-noho is similar to SNE. HGAR-noho and HGAR-noatt perform better than the baselines and LoanFeature, and the binary role properties prove to be effective when the guarantor and guarantee are treated differently in the embedding methods. The results of HGAR-all are significantly better than the sub-model methods, proving that high-order and attention layers could improve the model’s performance. Moreover, compared with other baselines, the improvement in results using HGAR-all is especially noticeable.

### 4.3 Effects of Attentional High-order Models

In this experiment, we report the precision of the top k nodes in delinquent prediction. In order to prove the effects of each module on the proposed methods, we experimented on the benchmark dataset with LoanFeature, HGAR-nolf, HGAR-noho, HGAR-noatt, and HGAR-all. Figure 1 shows the precision@k of default prediction with different k. The performance of HGAR-noho is similar to HGAR-noatt, and both are much better than HGAR-nolf and LoanFeature, showing the necessity of integrating topological and loan behavior features. HGAR-all performs much better than the four other baselines. The experimental results demonstrate the effectiveness of our proposed methods in loan risk assessment and default prediction.

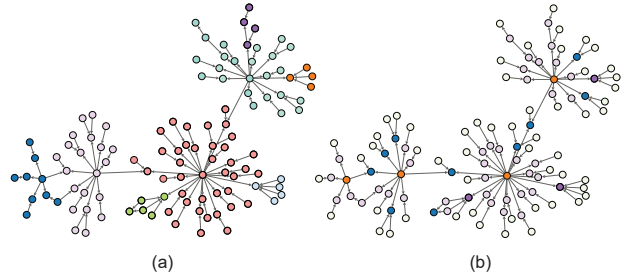


Figure 3: Case study: network visualization of GN75 (a) community detection by random walk; (b) clustered by HGAR.

### 4.4 Deployment and Case Studies

In HGAR, the attention layer is used to determine the importance of nodes during embedding within a certain time window. Figure 2 visualizes the attention weights of nodes in sampled time windows. We observe that a node with around 4 adjacent nodes is generally the most important since the colors of its heat distribution are brighter overall.

In order to validate the above observation, we employ an empirical study on a typical graph that is coded as GN75, which includes 112 nodes and 126 edges. We apply the K-Means algorithm on the learned embeddings of HGAR, as presented in Figure 3 (b). For comparison, we utilize random walk [Pons and Latapy, 2005] as the baseline for community detection, as presented in Figure 3 (a). It is clear that the results of random walk could only separate community equivalence. Our proposed method preserves better structure information in the learned embeddings, and the nodes with high attentions are clustered together, colored as blue and lilac. These companies show medium scale input, no output, star structures. This is because companies facing default need guarantors, but cannot gain many of them so as to form a normal star structure and do not have the ability to guarantee other companies. This phenomenon is also reported in [Meng *et al.*, 2017], which proves the observation from the attention layer visualization.

We further investigate the attention selected high default patterns during the deployment period. Specifically, we apply motif detection from high attention nodes for risk pattern discovery: (1) we employ the gtrieScanner approach [Ribeiro and Silva, 2012] on around 4 nodes motif detection, as nodes with around 4 neighbors are highlighted in Figure 2. (2) We match the motifs with the entire network and calculate the ratio for default firms as shown in Figure 4. (3) We rank de-

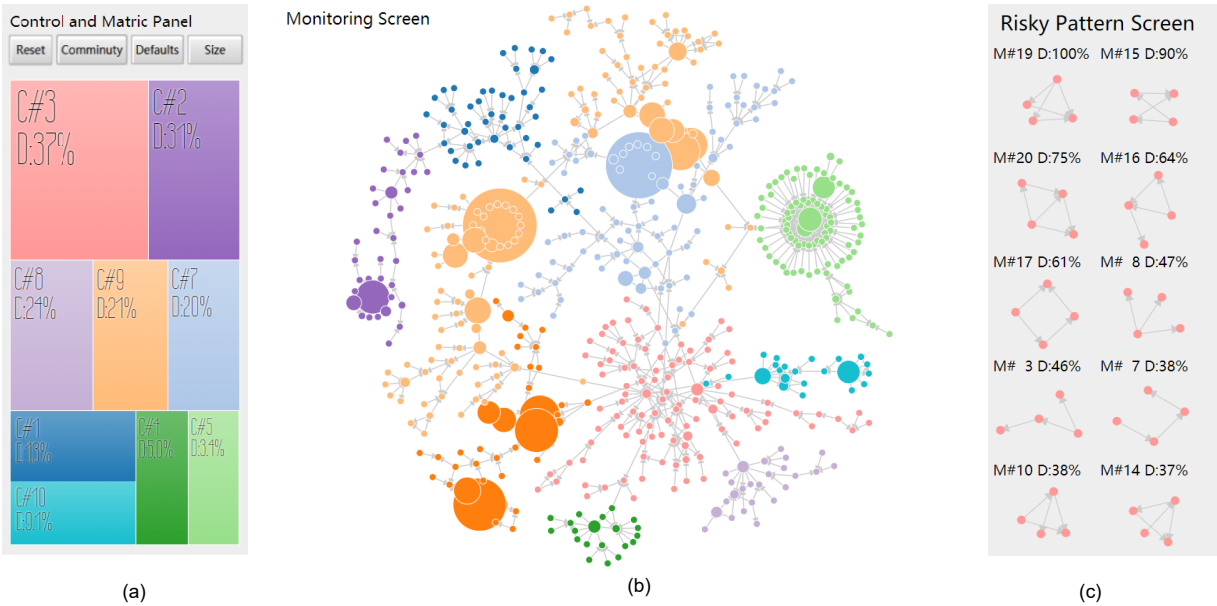


Figure 4: Deployed user interfaces of the loan management system, where “D:” denotes the defaults ratio. (a) presents the control and metric panel, including the risk statistics of each of the loan communities and control menus. (b) displays the loan status monitoring screen. The node size indicates the predicted delinquent probability, which is dynamic and changes with the time windows. Thus, risk managers could focus on risky and dominant companies. (c) lists all risky patterns discovered by the HGAR attention mechanism; our system could automatically search the whole complex network to find other company candidates that match selected risky patterns.

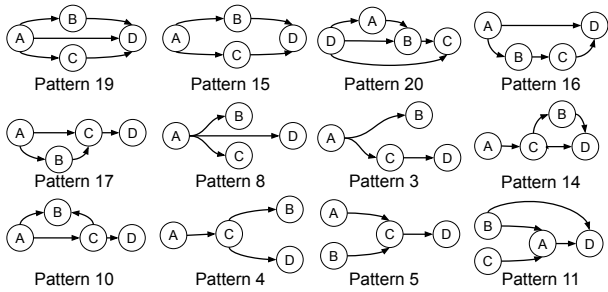


Figure 5: Risky patterns (4-vertex-motif structures) detected from GN32. Among them, patterns 15, 16, and 17 show single-input, single-output, and feed-forward structures.

tected patterns in descending default order, and mark the top  $n$  of them as high default pattern candidates. It is computationally expensive to detect all those motifs in the whole network, so we start from motifs of 4 vertices.

We select a typical graph coded as GN32, which consists of 103 enterprises; 36% of them default on 85% loans from the bank. Figure 5 shows the risky pattern candidates detected from GN32. As can be seen, although there are nearly 199 kinds of 4-vertex motif shapes theoretically, only 12 of them are discovered as high default pattern candidates. The structures of them are normally complex. It should be noted that some of these motifs are well known by domain experts. Taking motif 16 for example, it is a joint liability loan and motif 5 is a combination of a joint liability guarantee loan with a single guarantee. The top five risk motifs show single input, single output, feed forward structures.

Our method is deployed in the loan management system.

Figure 4 presents the system interface and main components. During the observation time window, it successfully warns 76% of true positive ratio of all risky loans with an acceptable 26% false positive rate, involving 2538 SMEs and 7006 guarantors. It is worth noting that these risky loans *failed* to be discovered by conventional scorecard based risk control system.

### 5 Conclusions

In the loan risk assessment systems, a company’s basic profile may not be completely accurate (it may be out of date or even include fake data), but the guarantee network provides a new dimension for loan risk assessment, which is more *trust-worthy*. Thus, in this paper, we present a novel loan risk prediction system using high-order graph attention representations. In particular, it contains three characteristics: 1) binary roles of nodes; 2) high-order adjacent approximation; 3) attentional layer selection. Extensive experiments demonstrate that our proposed methods are better than eight other state-of-the-art baseline methods. By observing gradual changes of attention weights after deployment, we discover risk guarantee patterns, which have been employed by bank’s internal risk managers to monitor possible delinquent companies and to curb systemic loan crises.

### Acknowledgments

The work was supported by the National Basic Research Program of China (2015CB856004), the Key Basic Research Program of Shanghai Science and Technology Commission, China (15JC1400103, 16JC1402800), and China Postdoctoral Science Foundation.

## References

- [Ahmed *et al.*, 2013] Amr Ahmed, Nino Shervashidze, Shравan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 37–48. ACM, 2013.
- [Allen and Babus, 2008] Franklin Allen and Ana Babus. Networks in finance. 2008.
- [Biggs, 2002] Tyler Biggs. Is small beautiful and worthy of subsidy? literature review. *International Finance Corporation (IFC). Washington, DC*, 2002.
- [Bougheas and Kirman, 2015] Spiros Bougheas and Alan Kirman. Complex financial networks and systemic risk: A review. In *Complexity and Geographical Economics*, pages 115–139. Springer, 2015.
- [Bravo *et al.*, 2015] Cristián Bravo, Lyn C Thomas, and Richard Weber. Improving credit scoring by differentiating defaulter behaviour. *Journal of the Operational Research Society*, 66(5):771–781, 2015.
- [Cheng *et al.*, 2018] Dawei Cheng, Zhibin Niu, Yi Tu, and Liqing Zhang. Prediction defaults for networked-guarantee loans. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 361–366. IEEE, 2018.
- [Cheng *et al.*, 2019] Da-Wei Cheng, Yi Tu, Zhen-Wei Ma, Zhi-Bin Niu, and Li-Qing Zhang. Bhonem: Binary high-order network embedding methods for networked-guarantee loans. *Journal of Computer Science and Technology*, 3(34):657–669, 2019.
- [Fischer and Molenaar, 2012] Gerhard H Fischer and Ivo W Molenaar. *Rasch models: Foundations, recent developments, and applications*. Springer Science & Business Media, 2012.
- [Fitzpatrick and Mues, 2016] Trevor Fitzpatrick and Christophe Mues. An empirical comparison of classification algorithms for mortgage default prediction: evidence from a distressed mortgage market. *European Journal of Operational Research*, 249(2):427–439, 2016.
- [Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.
- [Huang *et al.*, 2017] Xiao Huang, Jundong Li, and Xia Hu. Accelerated attributed network embedding. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 633–641. SIAM, 2017.
- [Kompass, 2007] Raul Kompass. A generalized divergence measure for nonnegative matrix factorization. *Neural computation*, 19(3):780–791, 2007.
- [Li *et al.*, 2014] Aaron Q Li, Amr Ahmed, Sujith Ravi, and Alexander J Smola. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 891–900. ACM, 2014.
- [Li *et al.*, 2018] Changsheng Li, Xiangfeng Wang, Weishan Dong, Junchi Yan, Qingshan Liu, and Hongyuan Zha. Joint active learning with feature selection via cur matrix decomposition. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [Liao *et al.*, 2018] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Attributed social network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2257–2270, 2018.
- [Meng *et al.*, 2017] Xiangfeng Meng, Yunhai Tong, Xinhai Liu, Yiren Chen, and Shaohua Tan. Netrating: Credit risk evaluation for loan guarantee chain in china. In *Pacific-Asia Workshop on Intelligence and Security Informatics*, pages 99–108. Springer, 2017.
- [Niu *et al.*, 2018] Zhibin Niu, Dawei Cheng, Liqing Zhang, and Jiawan Zhang. Visual analytics for networked-guarantee loans risk management. In *Pacific Visualization Symposium (PacificVis), 2018 IEEE*, pages 160–169. IEEE, 2018.
- [Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.
- [Pons and Latapy, 2005] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *International symposium on computer and information sciences*, pages 284–293. Springer, 2005.
- [Ribeiro and Silva, 2012] Pedro Ribeiro and Fernando Silva. Querying subgraph sets with g-tries. In *Proceedings of the 2nd ACM SIGMOD Workshop on Databases and Social Networks*, pages 25–30. ACM, 2012.
- [Siddiqi, 2012] Naeem Siddiqi. *Credit risk scorecards: developing and implementing intelligent credit scoring*, volume 3. John Wiley & Sons, 2012.
- [Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.
- [Van Vlasselaer *et al.*, 2013] Véronique Van Vlasselaer, Jan Meskens, Dries Van Dromme, and Bart Baesens. Using social network knowledge for detecting spider constructions in social security fraud. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 813–820. ACM, 2013.
- [Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.
- [Wang *et al.*, 2017] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *AAAI*, pages 203–209, 2017.