Computing and Informatics, Vol. 42, 2023, 257–279, doi: 10.31577/cai_2023_2_257

ASIAM-HGNN: AUTOMATIC SELECTION AND INTERPRETABLE AGGREGATION OF META-PATH INSTANCES FOR HETEROGENEOUS GRAPH NEURAL NETWORK

Xiaojun Lou

Department of School of Mathematics and Computer
Zhejiang A & F University
Hangzhou 311300, China
e-mail: 1932707397@qq.com

Guanjun Liu

Department of Computer Science
Key Laboratory of Embedded System and Service Computing
Tongji University
Shanghai 201804, China
e-mail: liuguanjun@tongji.edu.cn

Jian Li*

Department of School of Mathematics and Computer
Zhejiang A & F University
Hangzhou 311300, China
e-mail: lijian0120@foxmail.com

Abstract. In heterogeneous information network (HIN)-based applications, the existing methods usually use Heterogeneous Graph Neural Networks (HGNN) to handle some complex tasks. However, these methods still have some shortcomings: 1) they manually pre-select some meta-paths and thus some important ones are

* Corresponding author

missing, while the missing ones still contains the information and features of the node in the entire graph structure; and 2) they have no high interpretability since they do not consider the logical sequences in an HIN. In order to deal with them, we propose ASIAM-HGNN: a heterogeneous graph neural network combined with the automatic selection and interpretable aggregation of meta-path instances. Our model can automatically filter important meta paths for each node, while preserving the logical sequence between nodes, so as to solve the problems existing in other models. A group of experiments are conducted on real-world datasets, and the results demonstrate that the models learned by our method have a better performance in most of task scenarios.

Keywords: Graph neural networks, meta-paths, network representation learning, heterogeneous graph

1 INTRODUCTION

A Heterogeneous Information Network (HIN) defines a group of entities and their relations, and this heterogeneous representation can describe the real world more precisely compared to those homogeneous graphs. The emergence of HIN has triggered new explorations in many application scenarios such as relationship prediction [1, 2], recommendation [3, 4] and node classification [5].

However, because manual feature selection methods are generally used in these scenarios, it is hard for them to precisely express the characteristics of nodes. Although, some methods are proposed which can automatically mine node representation from a graph structure, such as Deepwalk [6], Metapath2vec [7] and ASNE [8], their ability is still limited when rich neighborhood information is expected to capture [9]. To solve this problem, some graph neural network methods are presented, such as: GCN [10], GAT [11] and GraphSAGE [12]. They can make good use of the feature feedback of adjacent nodes, and the aggregation embedding function is more powerful. However, they aim at homogenous graphs rather than heterogeneous ones. Consequently, they cannot distinguish the difference of node and edge attributes so that they cannot obtain a good performance for heterogeneous graphs [13, 9].

For overcoming this defect, some heterogeneous graph neural networks have been developed which mainly fall into two categories. The first one employs meta-paths to split a heterogeneous graph structure, and then to represent the node embeddings based on the extracted meta-path instances, e.g. HAN [9], HAHE [14], DeepHGNN [15] and MAGNN [16]. Another one such as HetGNN [13] does not use meta-path, but they usually apply the random walk algorithm to dig out the node embedding. However, they still have some problems. In the first category, they artificially preselect the types of meta-paths which influence the effect of the trained model, but this process is not interpretable because of the high subjectivity of manual selection. Additionally, they are inevitable to ignore the influence of unselected meta-paths of each node. In the second category, HetGNN [13] disorders the nodes selected by the random walk, and inputs them into a neural network. Obviously, it cannot maintain the logical sequence in the graph structure so that it does not have high interpretability.

Focusing on these problems, we propose ASIAM-HGNN, a method of Automatic Selection and Interpretable Aggregation of Meta-path instances used in Heterogeneous Graph Neural Network. ASIAM-HGNN can adaptively find "strongly correlated" meta-paths corresponding to nodes, and at the same time preserve the sequence relationship of the graph structure in the process of aggregating meta-paths, which solves:

1. The subjectivity caused by the artificial selection of meta-paths in existing models and Loss of information;
2. Low interpretability due to the loss of the original structure of the graph after random walk.

Specifically, we wander from each node by the random walk, and then get k "end-to-end" (the types of nodes at both ends of meta-path instances are the same) meta-path instances with the highest frequency from each node. In this way, we can collect the "strongly associated" meta-path instances for each node while preserving the connectivity and heterogeneity of a HIN. Meanwhile, our method can find out them automatically instead of manually. Additionally, the structural information retained in the meta-path is also saved in the embedded information when it passes through bi-LSTM, which solves the previous problem of weak interpretation. At last, k embedded path instances are aggregated through the attention layer which can make our model learn the influence factors of different meta-path instances on the target node, and help us express the accurate expression of the node in the graph structure. In the process of training, according to the triples corresponding to various types of nodes, our loss value calculation is based on all types of nodes, so the parameters optimized by the loss function will better reflect the real network situation. Through a group of experiments with real-world data, ASIAM-HGNN has a good effect in most task scenarios.

Our contributions can be summarized as follows:

- We employ a random walk-based approach to automatically search for "strongly correlated" meta-path instances. Our method can automatically and inclusively select all strongly associated meta-path instances around the node. Compared with artificial meta-path selection, our method is more objective and scientific, and will not cause information loss caused by artificial selection of meta-paths.
- We convert the node and meta-path instances of HIN into words and sentences respectively, and finally learn their embedding representations via bi-LSTM. Through the objective relationship sequence of "meta-path" and the learning of these sequence by bi-LSTM, we can preserve the original structure of graph data and the process of information transfer to the greatest extent on the basis of random walk, so as to solve the problem of interpretability low problem.

- We apply the attention mechanism to learn the attention coefficients of different nodes for different meta-path instances, and aggregate them together. In this process, multiple meta-path instances corresponding to the target node (including different meta path instances under the same meta path and meta path instances under different meta paths) are aggregated together. We learn the different effects of different meta-path instances corresponding to each node through the attention mechanism. Through this method, we try to find a more accurate representation for each node, and experiments prove that our model has a great performance.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 introduces some key definitions we used in this paper. Section 4 shows our Heterogeneous Graph Neural Network model in details. Section 5 demonstrates our model's performance and compares our model with the baselines. Section 6 indicates the result of Ablation Study. Section 7 summarizes our work.

2 RELATED WORK

The related work including 3 parts:

1. Graph Network Embedding,
2. Homogeneous GCNs,
3. Heterogeneous GCNs.

2.1 Graph Network Embedding

Graph network embedding is developed to extract the embedded information of nodes from the graph structure, so as to use this information for downstream tasks. For example, there are methods based on random walk and deep learning network, but these methods are all aimed at homogeneous graph networks [9].

For heterogeneous graph network, many methods have also been proposed by researchers. ESim [17] is proposed to use the pre-selected meta-path to learn graph structure. EOE [18], PTE [19] and HEER [20] have processed heterogeneous graph into several bipartite graphs. Then, LINE [21] is employed to learn the representation by preserving the first-order or the second-order proximities. SHNE [22] and Metapath2Vec [7] both employ the improved Skip-Gram model to learn representation of the node after processing the graph structure. HERec [23] and MCRec [3] are recommending models for heterogeneous graph. HERec employs the method of type restriction to seize semantic information in heterogeneous graph structures, while MCRec adopts convolutional neural network (CNN) to get paths and then trains them in the form of constructing triples (user, item, meta-path). However, none of these models consider the influence of all meta-path instances on different nodes. Additionally, above-methods require artificially pre-selection of multiple meta-paths.

2.2 Homogeneous GCNs

Due to the achievements of CNN in the field of image processing, many researchers also apply the idea of convolution to graph structure, forming homogeneous graph convolutional neural network. Homogeneous graph convolution models are roughly divided into two categories. The first one is in the spectral domain, Bruna et al. [24] propose a graph convolution method based on graph Laplace transform for the first time. Chebnet [25] applies k-order Chebyshev Polynomials, and on this basis, Kipf et al. [10] employ localized first-order approximation to design a graph model called graph convolutional network, which also achieves good performance. The second one is the spatial approach, Veličković et al. propose GAT [11], they employ the attention mechanism to learn the influence coefficients, which is the weights between different nodes, so as to aggregate nodes. Hamilton et al. propose GraphSAGE [12], which samples neighbor nodes to aggregate the target. However, these models are aim at homogeneous network. In the face of complex node and edge relations, they cannot distinguish them. Therefore, these models have not received good results in dealing with the information of heterostructures [9].

2.3 Heterogeneous GCNs

In recent years, due to the poor performance of several traditional graph neural network models on heterogeneous graph, many heterogeneous graph neural network models have emerged. The methods, such as: HAN [9], HAHE [14], DeepHGNN [15] and GraphInception [5], decompose heterogeneous graphs into multiple homogeneous graphs through different type of meta-paths, then use homograph neural network to conduct convolution aggregation. MAGNN [16] first proposes the internal aggregation of meta-paths. It proposes an aggregator to aggregate the information of meta-paths, and then uses attention mechanism to perform node aggregation. HIN-DRL [26] proposes a method of dynamic acquisition of sequence through meta-path and timestamp, then use skip-gram to learn the sequence's representation. HetGNN [13] first employs random walk with restart to sample "strongly associated" nodes on the heterogeneous graph, then aggregated nodes of various types, and finally aggregates different types through the attention mechanism to obtain the final representation of nodes. RANCH [27] and HIN2Grid [28] both combine the graph attention network with the convolutional neural network for more accurate embedding learning. RGCN [29] first performs in-type aggregation, and then aggregates according to the type of the edge. MBRep [30] learns triangle motif embedding in the graph structure to get the representation of the nodes. [31] proposes a model to calculate the entropy between different meta-paths. What HetSANN [32] and HGT [33] do was to directly apply the GAT method to the heterogeneous graph structure, calculate the attention coefficient between each node, and then perform aggregation according to the attention coefficient. However, compared with our model [9, 14, 15, 5, 16, 26, 33, 31] needs to manually select meta-paths, which is prone to subjectivity, and may miss some meta-paths that also contain important information. [13, 28, 27, 32] loses the objective sequence of the graph structure, which lead to low interpretability. [28, 30, 29, 32] does not take into account the structure of the meta-path, resulting in the information features contained in the meta-path being ignored.

3 PRELIMINARIES

We list three key definitions employed in our work for readability, which are from [34].

Definition 1 (HIN)**.** A heterogeneous information network (HIN) is a huge network with complex node types and relationships. A HIN is denoted as $G = (\mathcal{V}, \mathcal{E}, \phi, \psi, \mathcal{A}, \mathcal{R})$ where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ represents the feature set, $\mathcal{A}$ and $\mathcal{R}$ denote the sets of node and edge types such that $|\mathcal{A}| + |\mathcal{R}| > 2$, $\phi : \mathcal{V} \to \mathcal{A}$ and $\psi : \mathcal{E} \to \mathcal{R}$ are object type mapping function and link type mapping function.

Definition 2 (Meta-path)**.** A meta-path P is a path structure defined on the heterogeneous network and is represented in the form of $P_1 \xrightarrow{L_1} P_2 \xrightarrow{L_2} \dots \xrightarrow{L_l} P_{l+1}$, which defines a relationship collection $L = L_1 \circ L_2 \circ \cdots \circ L_l$ between types P_1 and P_{l+1}, and intermediate nodes on the meta-path are connected by these relationships.

Definition 3 (Meta-path instance)**.** Given a graph G and a meta-path set M, each specific path instance $p \in M$ that conforms to meta-path structure is called meta-path instance.

4 MODEL ASIAM-HGNN

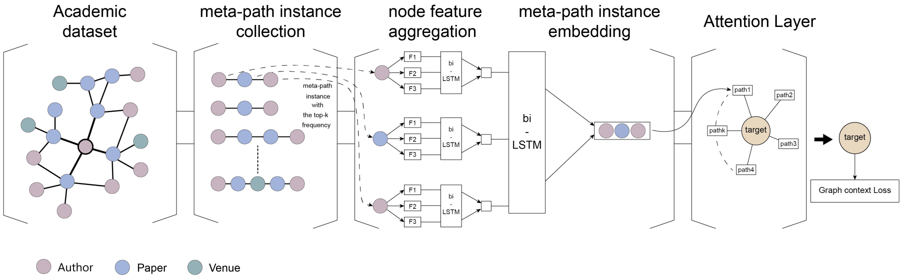


Figure 1. An overview of ASIAM-HGNN (Take academic data sets for example). ASIAM-HGNN consists of four parts: a) Meta-path instances collection, b) The aggregation of node features, c) Aggregation of the meta-path instances, d) Attention layer.

In this section, we will introduce four parts of our ASIAM-HGNN: Meta-path Instances Collection (collecting different meta-path instances for a group of target nodes and select the most correlated instance set), Node Features Aggregation (aggregating different types of node features), Meta-path Instance Embedding (converting meta-path a instance to a vector representation), Attention Layer (learning the influence factors of different meta-paths on the target node and aggregating them).

4.1 Meta-Path Instances Collection

In a complicated heterogeneous network structure, each node has a large number of neighboring nodes and associated edges from which the node's embedded information come. For mining the structural information around each node, we design a random walk based method that can adaptively find the "strongly associated" meta-path instances for each node and can dig out the node embedded information hidden in the graph structure. Meanwhile, we do not need to pre-select multiple types of meta-paths. The algorithm mainly contains the following two steps:

Step 1: Collecting meta-path instances for each node. When we employ random walk to reach a node whose type is the same as the target node's, we record this meta-path instance and then continue to start a new random walk from the target node. In order to ensure that the collected meta-path instances can accurately express the spatial and structural information of the target node, we will repeat this step many times. For example, for the Academic-II dataset, we have to repeat this step 100 times for each node, which means that 100 meta-path instances are stored for each node.

Step 2: Selecting instances with higher relevance from the collected meta-path instances. For the meta-path instances of a node collected by step one, we will select the k path instances with the highest occurrence frequency as the "strongly associated" meta-path instances of the node.

This strategy can solve the aforementioned defects, because it has the following two characteristics:

1. Intermediate nodes will be stored during the wandering process, the information of this part will not be discarded, this characteristic preserves the order and structure of the network to the greatest extent, so that the structural information of the network can be better mined;
2. Each random walk starts from the target node, so that all the information around the target node in the graph structure can be obtained to the greatest extent.

So random walk method can automatically select the meta-path instances that are "strongly associated" with around the target node, and the selected meta-path instances of each node are different which also preserves the heterogeneity of the network structure, this characteristic ensures that the information mined by our method is more rich and heterogeneous compared to the existing models.

Figure 2 shows the process of selecting meta path instances for target nodes under academic dataset. From the figure, we can clearly see that through our

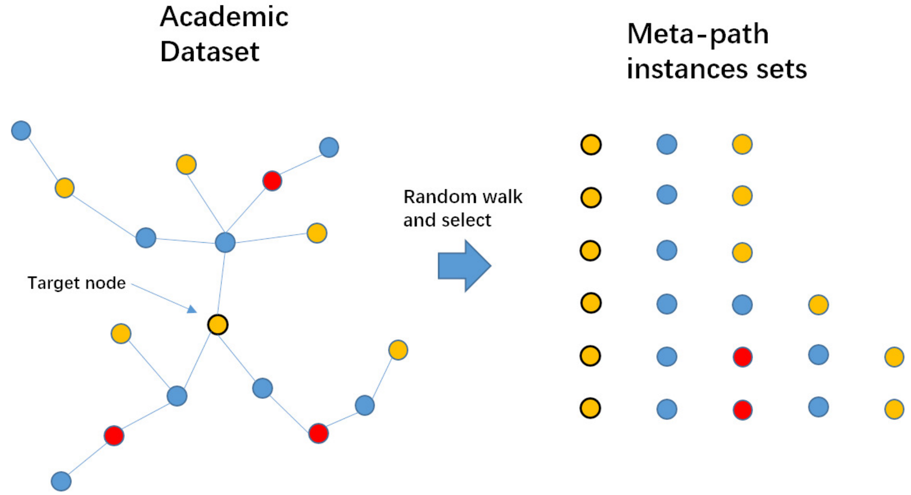


Figure 2. The example of meta-path instances selection for target nodes

selection of meta-path instances, most instances close to the target node are selected, including different meta-path and different instances under the same meta-path. In this way, we can find appropriate Metapath instances for each target node according to the original graph structure. Then we aggregate these meta-path instances in the same layer, which is helpful for our model to learn the different effects of different meta-path instances on each node.

4.2 Aggregation of Node Features

Given a meta-path instance p, it may contains different type of nodes, each type of node has different features and thus their dimensions are also different, which means that different type of nodes are in different semantic spaces. Therefore, we need some methods to aggregate their features and unify their dimensions. Here, we follow the method in the HetGNN [13].

Specifically, we first preprocess features of nodes in the network. For example, we use CNN to process image features, use ParVec [35] to process text features, change the feature dimensions to 128 dimensions, store these features in a three-dimensional matrix F, and then use the bi-LSTM network aggregating these features to obtain a node feature vector with a length dimension of 128 dimensions. This step can be understood as mapping the nodes with different lengths and different features to the same semantic spaces. The formula can be expressed as:

$$f(v) = \frac{\sum_{n \in C(n)} \left[\overrightarrow{LSTM} \left\{ f_1(n) \right\} \| \overleftarrow{LSTM} \left\{ f_1(n) \right\} \right]}{|C(n)|}, \tag{1}$$

where $C(n)$ denotes a set of node types, $f_1(n)$ represents the features of node n. The calculation process of $LSTM$ can be expressed as:

$$\begin{aligned}
y_i &= \sigma\left(\mu_y f\left(x_i\right) + w_y h_{i-1} + b_y\right), \\
q_i &= \sigma\left(\mu_q f\left(x_i\right) + w_q h_{i-1} + b_q\right), \\
e_i &= \sigma\left(\mu_e f\left(x_i\right) + w_e h_{i-1} + b_e\right), \\
\hat{d}_i &= \tanh\left(\mu_d f\left(x_i\right) + w_d h_{i-1} + b_d\right), \\
d_i &= q_i \circ d_{i-1} + y_i \circ \hat{d}_i, \\
j_i &= \tanh\left(d_i\right) \circ e_i,
\end{aligned} \tag{2}$$

where $\circ$ denotes the Hadamard product, y_i, q_i and e_i are the vectors of forget gate, b is the parameter that need to learn through the network, and j_i denotes the hidden output of the i^{th} unit Refering to HetGNN, this process can ensure a better effect on the feature aggregation of nodes.

4.3 Meta-Path Instance Embedding

Before dealing with meta-path instances, we must again emphasize the function of meta-path instances in this paper, which will help one understand our next work. We believe that the "end-to-end" meta-path instances in the heterogeneous network is responsible for most of the information connection and transmission. In the real world, the end-to-end information transfer method includes most of the information in the network, such as the relationship between authors in academic datasets, the relationship between articles, etc. At the same time, the methods in previous models can also prove this, for example, the HAN model directly discards the intermediate nodes of meta-path, only retains both ends' nodes, and finally achieves good performance. However, we find that information transfer in heterogeneous networks is bidirectional. For instance, in the Academic dataset, there is a meta-path instance $p(P^1 - V^1 - P^2)$ such that P^1 affects P^2 through V^1 to a certain extent, P^2 also affects P^1 in this way. We hope that this bidirectional information can also be reflected in the aggregation of meta-path instances. To this end, we choose bi-LSTM. This bidirectional LSTM network can train semantic information in two directions. Unlike HetGNN that inputs the LSTM after finding the nodes out of order, we retain the original structure of the heterogeneous network when inputting. The preservation of this natural structure makes our model more convincing.

The aggregation formula for the meta-path is:

$$\begin{aligned}
&n_1, n_2, n_3 \in P, \\
&\qquad F = \operatorname{concat}(f(n_1), f(n_2), f(n_3)), \\
&\mathit{embedding} = \left[\overrightarrow{\mathrm{LSTM}}\{F(i)\} \| \overleftarrow{\mathrm{LSTM}}\{F(i)\}\right],
\end{aligned} \tag{3}$$

where n_1, n_2, n_3 are the nodes that makes up the meta path p, $F(i)$ denotes the instance's vector which have concated different vector of nodes, and $||$ denotes concatenation.

When using bi-LSTM, we understand a meta-path instance as a sentence with semantic meaning and the nodes as the words in the sentence. Different from the previous aggregation of node vectors, the previous step is to average the output of each unit. When aggregating the meta-path, since we want to get the representation of this path, we take the output of the last unit of the meta-path sequence in the bi-LSTM network as the embedding representation for that path.

4.4 Attention Layer

After getting the vector representation of each path, we also need to aggregate them to get the final representation of the target node. At this time, we must consider the imbalance of the heterogeneous network, which means different meta-path instances have totally different influence on the target node. We employ an attention layer to complete this step of aggregation operation. Among them, the feature aggregation of node $Node_v$ is expressed as:

$$\gamma_v = a^{v,v} f_1(v) + \sum_{x \in Path_v} a^{v,x} f_2^x(v), \tag{4}$$

where $a^{v,x}$ denotes the importance between v and x, $f_1(v)$ is the embedded representation of target node which is calculated in the Section 4.2, $f_2^x(v)$ is the embedded representation of meta-path instance which is aggregated in the Section 4.3.

The calculation process of the attention coefficient a between the node and the meta-path is:

$$a^{v,x} = \frac{\exp\left\{ \mathit{LeakyRelu}\left(u^T \left[f_i \| f_1(x)\right]\right)\right\}}{\sum_{f_j \in Path} \exp\left\{ \mathit{LeakyRelu}\left(u^T \left[f_j \| f_1(x)\right]\right)\right\}}, \tag{5}$$

where $Path$ represents the path instance set corresponding to the node, and u is a learnable parameter.

4.5 Loss Function

Since most of the data in many real-world data sets are not labeled, in view of this situation, we think that the unsupervised training method can better meet the needs of reality. We collect a certain number of triples by negative sampling [36]. Then, based on those triples, we can optimize the parameters and weights of the model according to the following loss function:

$$\mathcal{L} = \sum_{(n, n_v, n_{v'}) \in triples} \log \sigma\left(\mathcal{E}_{n_v} \cdot \mathcal{E}_n\right) + \log \sigma\left(-\mathcal{E}_{n_{v'}} \cdot \mathcal{E}_n\right), \tag{6}$$

where v denotes the target, n_v denotes the positive sample, $n_{v'}$ denotes the negative sample, σ represents the sigmoid function, $\mathcal{E}_n$ is the calculated embed of target node n.

Algorithm 1: Training process of ASIAM-HGNN

```
Input: pre-trained content features of nodes $n \in N$
trpiples sets $T(n, n_{pos}, n_{neg})$
meta-path instaces sets $M$ for each nodes
Output: node embeddings
while not done do
    calculate mixed content features of nodes $n \in N$ by Equation (1)
    learn embeddings of each meta-path instances in sets $M$ by Equation (3)
    Aggregated mutiple meta-path instances through the attention layer to
      obtain the embedded representation of nodes by Equation (5)
    Compute loss by sending learned node embeddings and trpiples sets
      $T(n, n_{pos}, n_{neg})$ into Equation (6)
    update parameters by Adam
end
```

5 EXPERIMENT

We do experiments to evaluate the performance of ASIAM-HGNN on different tasks (link prediction and node classification).

5.1 Dataset

Academic: We follow the preprocessed method of HetGNN [13] to deal with the Academic Heterogeneous graph datasets from the Aminer database [37], and the detailed structure of the data is visible in Table 1.

DBLP: We used the DBLP dataset processed by [9], we use the BOW representations of keywords as the content features of the author nodes, and the detailed structure of the data is visible in Table 1.

IMDB: It is an online movie dataset, including three types of nodes: movie, actor and director, We use the BOW representations of plots as the content features of the movies, and the detailed structure of the data is visible in Table 1.

5.2 Baseline

We use six representative graph structure models as the baseline.

Dataset	Node Type	Edge Type
Academic-I	Paper:16073 Author:111409 Venue150	A–P P–P P–V
Academic-II	Paper:28646 Author:21044 Venue:18	A–P P–P P–V
DBLP	Paper:14328 Author:4057 Venue:20	A–P P–V
IMDB	Movie:3627 director:1714 actor:4340	M–D M–A

These are all Heterogeneous information Network.

Table 1. Dataset

Metapath2vec (mp2vec) [7]: This is a very advanced graph embedding model which applies the random walk method to graph structures most early.

ASNE [8]: This is an attribute graph embedding model that learns node features by combining "latent" features and attribute features.

SHNE [22]: This is also an attribute graph model, which learns node embedding by combining the tightness of the graph and semantic relevance.

GraphSAGE [12]: It is a neural network model that obtains node characteristics by sampling surrounding nodes and is widely used in application fields.

GAT [11]: It learns the degree of influence between nodes through the attention mechanism, thereby obtaining node representation.

HetGNN [13]: It samples "strongly associated" nodes through random walks and obtains node embeddings through node aggregation, which have great performance on Heterogeneous structure.

HAN [9]: It divides the heterogeneous graph into multiple homogeneous graphs through meta-path, and then aggregates through the double-layer attention mechanism to obtain the final representation of the nodes.

MAGNN [16]: It is a heterogeneous graph neural network model, which supplements the node information in the middle of the meta-path on the basis of HAN.

For homogeneous graph neural networks such as [11] and [12], we unify the nodes in the heterogeneous graph as the degree of advancement, and then input the data as a homogeneous graph.

5.3 Link Prediction

Link prediction is to learn the composition and structure of the existing edges in the structure, and then use the learned information to judge the possibility of the existence of unknown edge relationships. This task is widely applied in many fields, so we will use it as the first task scenario. The evaluation indicators we have chosen are AUC and $F1$, which can be expressed by formula:

$$AUC = \frac{\sum_{ins_i \in \text{positive Class}} \text{rank}_{ins_i} - \frac{P(1+P)}{2}}{P \times N}, \tag{7}$$

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP+FP}, \\ \text{Recall} &= \frac{TP}{TP+FN}, \\ F1\text{score} &= \frac{2\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}, \end{aligned} \tag{8}$$

In this task, we employ the same evaluation method as [13]. Specially, the first step of link prediction is to apply our model to train the embedding representation of each node, the second step is to employ the original links in the structure to train the binary classifier and the third step utilizes this binary classifier to evaluate the equal number of non-link relationships. In this process, we will only evaluate new links in the training data, the link embedding between the node and node is obtained by multiplying the embeddings of the nodes at both ends.

In Table 2, the experimental results for this task are shown, with the best results for each experiment marked in bold. According to these data: our model is better than all benchmark models in link prediction between nodes which have same type (Author-Author). Compared with HetGNN (after random wandering, all types of nodes are out of order), our model also uses the random walk method to preprocess the graph structure, but on the basis of random walk, we retain the original graph structure by means of meta path. Random walk gives our structure strong adaptability (adaptively selecting meta path instances for each node). The structure of meta path maintains the process of information transmission to a certain extent, The experimental results also show that our effect is better than HetGNN.

The main reason is that we obtain the representation of nodes through the aggregation of meta-path instances, and both ends of the meta-path instances are nodes with the same type. This structure can help our model to learn the relationship between the same types of nodes in the heterogeneous information network. Because

Dataset	Metric	M2vec	ASNE	SHNE	GSAGE	GAT	HetG	ASIAM-HGNN
$A - I_{2003}$	AUC	0.636	0.683	0.696	0.694	0.701	0.714	**0.735**
	F1	0.435	0.584	0.597	0.586	0.606	0.620	**0.633**
$A - I_{2002}$	AUC	0.626	0.667	0.688	0.681	0.691	0.710	**0.733**
	F1	0.412	0.554	0.590	0.567	0.589	0.615	**0.631**
$A - II_{2013}$	AUC	0.596	0.689	0.683	0.695	0.678	0.717	**0.732**
	F1	0.348	0.643	0.639	0.615	0.613	0.669	**0.683**
$A - II_{2012}$	AUC	0.586	0.671	0.672	0.676	0.655	0.701	**0.724**
	F1	0.318	0.615	0.612	0.573	0.560	0.642	**0.659**

$A - I_{2012}$ denotes that we use the data before 2012 as the train set, namely, the data after 2012 is the test set.

10 % of the test set is divided into the validation set.

Table 2. Experimental results of link prediction

of this characteristic, we believe that our model can have a great performance in many fields such as:

1. searching for user relationships in social networks;
2. disease relationship prediction;
3. compound relationship prediction.

5.4 Node Classification

Node classification is to classify nodes according to the existing node characteristics and labels, thereby predicting the category of nodes which don't have labels. The indicators used to evaluate the classification effect are: Micro-F1 and Macro-F1, which can be expressed by formula:

$$\begin{aligned}\text{Precision}_{\text{micro}} &= \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} \text{TP}_i + \sum_{i=1}^{n} \text{FP}_i}, \\ \text{Recall}_{\text{micro}} &= \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} \text{TP}_i + \sum_{i=1}^{n} \text{FN}_i},\end{aligned} \tag{9}$$

$$F1_{\text{micro}} = 2 \cdot \frac{\text{Precision}_{\text{micro}} \cdot \text{Recall}_{\text{micro}}}{\text{Precision}_{\text{micro}} + \text{Recall}_{\text{micro}}}, \tag{10}$$

$$\begin{aligned}\text{Precision}_{\text{macro}} &= \frac{\sum_{x=1}^{m} \text{Precision}_x}{m}, \\ \text{Recall}_{\text{macro}} &= \frac{\sum_{x=1}^{m} \text{Recall}_x}{m},\end{aligned} \tag{11}$$

$$F1_{\text{macro}} = 2 \cdot \frac{\text{Precision}_{\text{macro}} \cdot \text{Recall}_{\text{macro}}}{\text{Precision}_{\text{macro}} + \text{Recall}_{\text{macro}}}, \tag{12}$$

where i is the category of the node, and Precision and Recall are calculated in the same way as $F1$.

In this task, we follow the mathod of GrapgSAGE [12]. The data set itself does not have labels, and if magazines are designated as labels, there are too many types of labels. Therefore, we divide journals into four broad categories based on their characteristics and publications, and our criteria for labeling authors are the areas in which most of his papers are published. We learn the node representation through the ASIAM-HGNN model, and then input node representation and label into the logistic regression model. The dataset is divided into training set and test set according to different ratios, and 10 % of the test set is divided into the validation set.

Dataset	Train (%)	Metrics	M2-VEC	ASNE	SHNE	GSAGE	GAT	HetG	HAN	MAGNN	ASIAM-HGNN
A-I	10	Mac-F1	0.972	0.965	0.939	0.978	0.962	0.978	0.971	0.975	**0.982**
		Mic-F1	0.973	0.967	0.941	0.978	0.963	0.978	0.972	0.975	**0.983**
	30	Mac-F1	0.975	0.969	0.939	0.979	0.965	0.981	0.972	0.976	**0.983**
		Mic-F1	0.975	0.970	0.941	0.980	0.965	**0.982**	0.973	0.976	**0.982**
DBLP	20	Mac-F1	0.815	–	0.825	0.845	0.847	0.908	0.894	0.893	**0.913**
		Mic-F1	0.817	–	0.827	0.846	0.849	0.908	0.891	0.907	**0.912**
	40	Mac-F1	0.827	–	0.831	0.856	0.854	0.917	0.907	0.909	**0.919**
		Mic-F1	0.826	–	0.831	0.862	0.861	0.919	0.902	0.913	**0.919**
IMDB	20	Mac-F1	0.421	–	–	0.523	0.515	0.537	0.521	0.521	**0.539**
		Mic-F1	0.434	–	–	0.520	0.517	0.531	0.521	0.531	**0.536**
	40	Mac-F1	0.433	–	–	0.526	0.519	0.551	0.534	0.541	**0.554**
		Mic-F1	0.442	–	–	0.527	0.520	**0.559**	0.535	0.549	0.555

Table 3. Experimental results of node classification

The data of the tasks are shown in Table 3. It is clear that Our model has the best effect on most datasets. The main reason is that Our model collects meta path instances sufficient to cover most of the structural information of nodes through random walk method. Meta path instances enhance the ability of our model to learn the relationship between nodes through end-to-end form. So our model can achieve better performance in node classification.

Figure 3 shows the visualization of the embedded vector of the node learned from our model. We reduce the dimension of the high-dimensional vector representation through the TSNE method. Through the visual picture, we can clearly see that our model divides the nodes into four obvious categories.

6 ANALYSIS

In this section, the impact of some hyper parameter and intermediate structures will be shown to demonstrate the stability of the model.

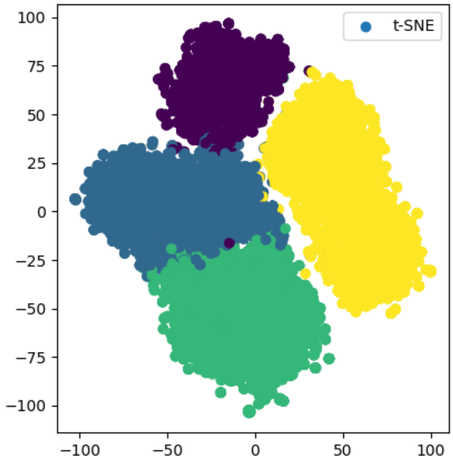


Figure 3. Embedded vector visualization in Academic-I dataset, each dot represents an author, and the color of the dot represents the corresponding field of the author

6.1 Ablation Study

In order to prove that the algorithm we selected in the model has a positive effect, we have done some ablation experiments and designed two different models for comparison.

1. We use the fully connected neural network MLP to embed the meta-path instances (ASIAM-MLP).
2. We use the fully connected neural network MLP to embed the meta-path instances (ASIAM-RNN).

The results of the ablation Study are shown in Figure 4, which can demonstrate:

1. ASIAM-HGNN uses various feature information of nodes which are very helpful to the improvement of experimental effects.
2. ASIAM-HGNN uses the bi-LSTM network in the method of learning the embedding representation of the meta-path instances.

In this way, we can not only preserve the objectively existing logical sequences in heterogeneous graphs, but also learn bi-directional information in meta-path instances through bi-LSTM.

6.2 Hyper-Parameters Sensitivity

Hyperparameters play an important role in our model, and they determine the amount of information we obtain. We design an experiment to analyze the impact

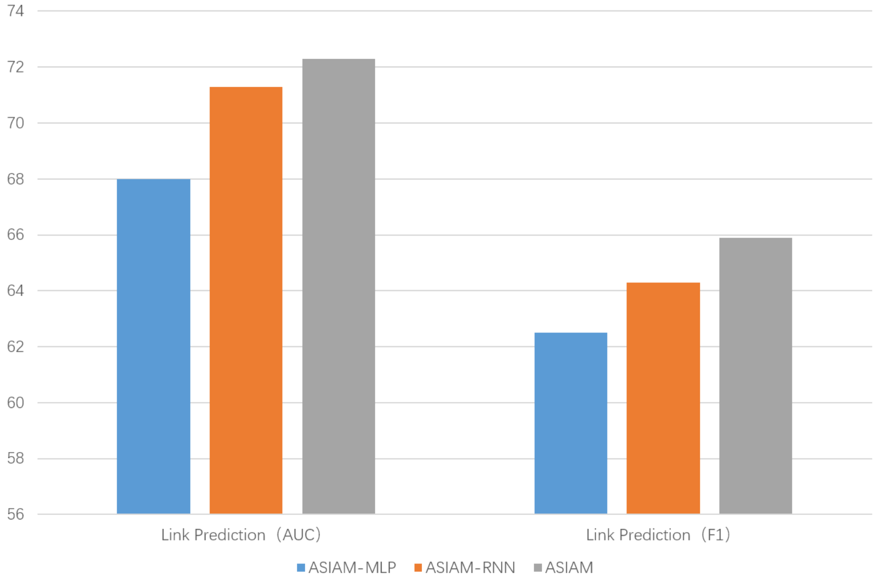


Figure 4. Performances of different models

of changes in the value of k (the number of meta-path instances we choose for each node) and dimensions of node embedding.

According to Figure 5, we can figure out that:

- With the change of k value, the experimental results will not change much, which shows the stability of our model.
- When the k value continues to rise from 3 to 8, the scores of AUC and $F1$ both reach the highest value. When k is equal to 6, there is a slight drop, which may be due to overfitting.

According to Figure 6, we can figure out that:

- When the dimension of the embedded vector increases step by step from 32 to 256, the scores of AUC and F1 indicators continue to improve. When the dimension reaches 128, the score reaches the highest point.

7 CONCLUSIONS

In our article, we propose a neural network model based on a heterogeneous graph structure to solve the previously mentioned problems of meta-path instances selection and low interpretability. We employ a meta-path instance selection method which based on random walk to automatically and inclusively select meta-path instances for each node, this step can automatically select the most associated meta-

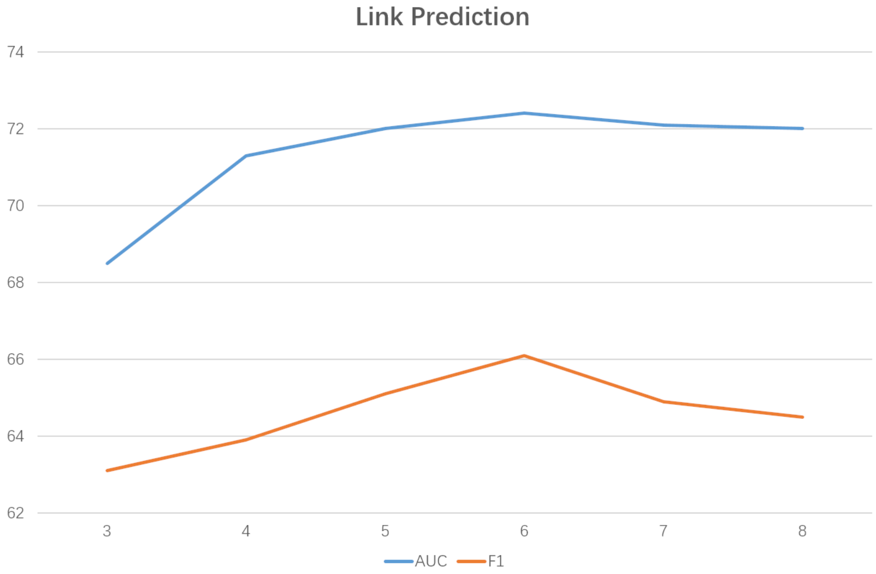


Figure 5. Performances of different value of k

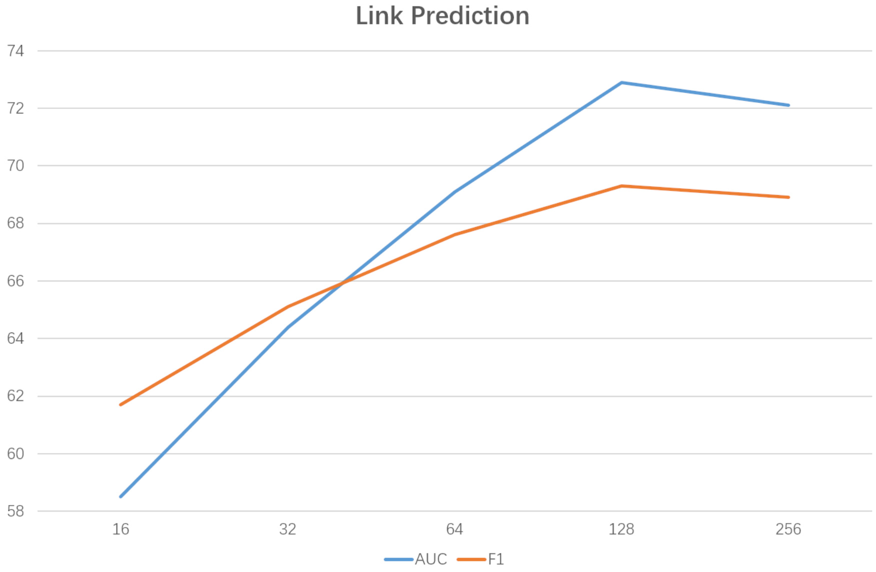


Figure 6. Performances of different value of embedding dimensions

path instances for each node because it makes full use of the objectively existing information in the graph structure. We take bi-LSTM for embedding learning of meta-path instance, which is able to preserves the logical sequence in the graph structure and earn higher interpretability. At the same time, our method aggregates different types of meta paths and single meta path instances of the same type of meta path at the same semantic level, which is conducive to learning the impact of different meta path instances on different nodes. This meta-path instances selection strategy refined to each node can more accurately learn the embedded representation of nodes. Our model has been tested on multiple tasks such as link prediction and node classification. In the task of link prediction, there is a 1.3 % to 2.3 % improvement over the baseline model. In the task of node classification, our model has a 1 % improvement over the baseline model on both the academic dataset and the DBLP dataset, the improvement is 0.5 %–1.8 % on the IMDB dataset. These experimental data confirm that our model has excellent performance in different application scenarios.

Acknowledgements

The paper is supported by the National Nature Science Foundation of China (No. 62-172299) and the Shanghai Science and Technology Committee (No. 22511105500).

REFERENCES

[1] Chen, T.—Sun, Y.: Task-Guided and Path-Augmented Heterogeneous Network Embedding for Author Identification. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, 2017, pp. 295–304, doi: 10.1145/3018661.3018735.

[2] Zhang, C.—Huang, C.—Yu, L.—Zhang, X.—Chawla, N. V.: Camel: Content-Aware and Meta-Path Augmented Metric Learning for Author Identification. Proceedings of the 2018 World Wide Web Conference, 2018, pp. 709–718, doi: 10.1145/3178876.3186152.

[3] Hu, B.—Shi, C.—Zhao, W. X.—Yu, P. S.: Leveraging Meta-Path Based Context for Top-N Recommendation with a Neural Co-Attention Model. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 1531–1540, doi: 10.1145/3219819.3219965.

[4] Ren, X.—Liu, J.—Yu, X.—Khandelwal, U.—Gu, Q.—Wang, L.—Han, J.: Cluscite: Effective Citation Recommendation by Information Network-Based Clustering. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 821–830, doi: 10.1145/2623330.2623630.

[5] Zhang, Y.—Xiong, Y.—Kong, X.—Li, S.—Mi, J.—Zhu, Y.: Deep Collective Classification in Heterogeneous Information Networks. Proceedings of the 2018 World Wide Web Conference, 2018, pp. 399–408, doi: 10.1145/3178876.3186106.

[6] Perozzi, B.—Al-Rfou, R.—Skiena, S.: Deepwalk: Online Learning of Social Representations. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710, doi: 10.1145/2623330.2623732.

[7] Dong, Y.—Chawla, N. V.—Swami, A.: Metapath2vec: Scalable Representation Learning for Heterogeneous Networks. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 135–144, doi: 10.1145/3097983.3098036.

[8] Liao, L.—He, X.—Zhang, H.—Chua, T. S.: Attributed Social Network Embedding. IEEE Transactions on Knowledge and Data Engineering, Vol. 30, 2018, No. 12, pp. 2257–2270, doi: 10.1109/TKDE.2018.2819980.

[9] Wang, X.—Ji, H.—Shi, C.—Wang, B.—Ye, Y.—Cui, P.—Yu, P. S.: Heterogeneous Graph Attention Network. The World Wide Web Conference, 2019, pp. 2022–2032, doi: 10.1145/3308558.3313562.

[10] Kipf, T. N.—Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. Arxiv Preprint Arxiv:1609.02907, 2016, doi: 10.48550/arXiv.1609.02907.

[11] Veličković, P.—Cucurull, G.—Casanova, A.—Romero, A.—Lio, P.—Bengio, Y.: Graph Attention Networks. Arxiv Preprint Arxiv:1710.10903, 2017, doi: 10.48550/arXiv.1710.10903.

[12] Hamilton, W. L.—Ying, R.—Leskovec, J.: Inductive Representation Learning on Large Graphs. Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 1025–1035, doi: 10.48550/arXiv.1706.02216.

[13] Zhang, C.—Song, D.—Huang, C.—Swami, A.—Chawla, N. V.: Heterogeneous Graph Neural Network. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019, pp. 793–803, doi: 10.1145/3292500.3330961.

[14] Zhou, S.—Bu, J.—Wang, X.—Chen, J.—Wang, C.: Hahe: Hierarchical Attentive Heterogeneous Information Network Embedding. Arxiv Preprint Arxiv:1902.01475, 2019, doi: 10.48550/arXiv.1902.01475.

[15] Wang, S.—Chen, Z.—Li, D.—Li, Z.—Tang, L. A.—Ni, J.—Rhee, J.—Chen, H.—Yu, P. S.: Attentional Heterogeneous Graph Neural Network: Application to Program Reidentification. Proceedings of the 2019 SIAM International Conference on Data Mining, SIAM, 2019, pp. 693–701, doi: 10.1137/1.9781611975673.78.

[16] Fu, X.—Zhang, J.—Meng, Z.—King, I.: Magnn: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. Proceedings of the Web Conference 2020, 2020, pp. 2331–2341, doi: 10.1145/3366423.3380297.

[17] Shang, J.—Qu, M.—Liu, J.—Kaplan, L. M.—Han, J.—Peng, J.: Meta-Path Guided Embedding for Similarity Search in Large-Scale Heterogeneous Information Networks. Arxiv Preprint Arxiv:1610.09769, 2016, doi: 10.48550/arXiv.1610.09769.

[18] Xu, L.—Wei, X.—Cao, J.—Yu, P. S.: Embedding of Embedding (EOE) Joint Embedding for Coupled Heterogeneous Networks. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, 2017, pp. 741–749, doi: 10.1145/3018661.3018723.

[19] Tang, J.—Qu, M.—Mei, Q.: Pte: Predictive Text Embedding Through Large-Scale Heterogeneous Text Networks. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1165–1174, doi: 10.1145/2783258.2783307.

[20] Shi, Y.—Zhu, Q.—Guo, F.—Zhang, C.—Han, J.: Easing Embedding Learning by Comprehensive Transcription of Heterogeneous Information Networks. Proceedings of the 24^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 2190–2199, doi: 10.1145/3219819.3220006.

[21] Tang, J.—Qu, M.—Wang, M.—Zhang, M.—Yan, J.—Mei, Q.: Line: Large-Scale Information Network Embedding. Proceedings of the 24^{th} International Conference on World Wide Web, 2015, pp. 1067–1077, doi: 10.1145/2736277.2741093.

[22] Zhang, C.—Swami, A.—Chawla, N. V.: Shne: Representation Learning for Semantic-Associated Heterogeneous Networks. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 690–698, doi: 10.1145/3289600.3291001.

[23] Shi, C.—Hu, B.—Zhao, W. X.—Philip, S. Y.: Heterogeneous Information Network Embedding for Recommendation. IEEE Transactions on Knowledge and Data Engineering, Vol. 31, 2018, No. 2, pp. 357–370, doi: 10.1109/TKDE.2018.2833443.

[24] Bruna, J.—Zaremba, W.—Szlam, A.—LeCun, Y.: Spectral Networks and Locally Connected Networks on Graphs. Arxiv Preprint Arxiv:1312.6203, 2013, doi: 10.48550/arXiv.1312.6203.

[25] Defferrard, M.—Bresson, X.—Vandergheynst, P.: Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. Advances in Neural Information Processing Systems, Vol. 29, 2016, pp. 3844–3852, doi: 10.48550/arXiv.1606.09375.

[26] Meilian, L.—Danna, Y.: HIN_DRL: A Random Walk Based Dynamic Network Representation Learning Method for Heterogeneous Information Networks. Expert Systems with Applications, Vol. 158, 2020, Art. No. 113427, doi: 10.1016/j.eswa.2020.113427.

[27] Tong, N.—Tang, Y.—Chen, B.—Xiong, L.: Representation Learning Using Attention Network and CNN for Heterogeneous Networks. Expert Systems with Applications, Vol. 185, 2021, Art. No. 115628, doi: 10.1016/j.eswa.2021.115628.

[28] Zhang, Z.—Chen, C.—Chang, Y.—Hu, W.—Zheng, Z.—Zhou, Y.—Sun, L.: HIN2Grid: A Disentangled CNN-Based Framework for Heterogeneous Network Learning. Expert Systems with Applications, Vol. 187, 2022, 115823 pp., doi: 10.1016/j.eswa.2021.115823.

[29] Schlichtkrull, M.—Kipf, T. N.—Bloem, P.—Van Den Berg, R.—Titov, I.—Welling, M.: Modeling Relational Data with Graph Convolutional Networks. The Semantic Web (ESWC 2018), Springer, 2018, pp. 593–607, doi: 10.1007/978-3-319-93417-4_38.

[30] Hu, Q.—Lin, F.—Wang, B.—Li, C.: MBRep: Motif-Based Representation Learning in Heterogeneous Networks. Expert Systems with Applications, Vol. 190, 2022, Art. No. 116031, doi: 10.1016/j.eswa.2021.116031.

[31] Molaei, S.—Farahbakhsh, R.—Salehi, M.—Crespi, N.: Identifying Influential

Nodes in Heterogeneous Networks. Expert Systems with Applications, Vol. 160, 2020, Art. No. 113580, doi: 10.1016/j.eswa.2020.113580.

[32] Hong, H.—Guo, H.—Lin, Y.—Yang, X.—Li, Z.—Ye, J.: An Attention-Based Graph Neural Network for Heterogeneous Structural Learning. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 4132–4139, doi: 10.1609/aaai.v34i04.5833.

[33] Hu, Z.—Dong, Y.—Wang, K.—Sun, Y.: Heterogeneous Graph Transformer. Proceedings of the Web Conference 2020, 2020, pp. 2704–2710, doi: 10.1145/3366423.3380027.

[34] 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, Vol. 4, 2011, No. 11, pp. 992–1003, doi: 10.14778/3402707.3402736.

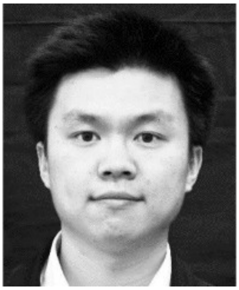
[35] Cebrian, J. M.—Jahre, M.—Natvig, L.: Parvec: Vectorizing the PARSEC Benchmark Suite. Computing, Vol. 97, 2015, No. 11, pp. 1077–1100, doi: 10.1007/s00607-015-0444-y.

[36] Mikolov, T.—Sutskever, I.—Chen, K.—Corrado, G. S.—Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality. Advances in Neural Information Processing Systems, Vol. 26, 2013, doi: 10.5555/2999792.2999959.

[37] Tang, J.—Zhang, J.—Yao, L.—Li, J.—Zhang, L.—Su, Z.: Arnetminer: Extraction and Mining of Academic Social Networks. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 990–998, doi: 10.1145/1401890.1402008.

Xiaojun Lou is a postgraduate in the School of Mathematics and Computer Science, at Zhejiang Agriculture and Forestry University. His current research interests include graph neural networks, recommender systems, and multimodality.

Guanjun Liu received his Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2011. He was a Post-Doctoral Research Fellow with the Singapore University of Technology and Design, Singapore, from 2011 to 2013, and a Post-Doctoral Research Fellow with the Humboldt University of Berlin, Berlin, Germany, from 2013 to 2014, supported by the Alexander von Humboldt Foundation. He is currently a Professor at the Department of Computer Science, at Tongji University. He has authored over 130 papers and 3 books. He served as a Guest Editor for some journals including the IEEE Transactions on Computational Social Systems, the Mathematical Problems in Engineering, and the Journal of Software (in Chinese), and served as a Program Committee Member for many international conferences. He is now an Associate Editor of the IEEE Transactions on Computational Social Systems. His research interests include Petri net theory, model checking, machine learning, cyber-physical systems, workflow, and credit card fraud detection.



Jian Li received his Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2020. He is currently a Vice Professor at the School of Mathematics and Computer Science, Zhejiang A & F University, Hangzhou, China. His research interests include matrix decomposition, graph neural networks, and their applications.