A Semantic Aware Meta-Path Model for Heterogeneous Network Representation Learning

Article *in* IEEE Access · December 2020

DOI: 10.1109/ACCESS.2020.3043269

4 authors:

Yiping Yang
Wuhan University
3 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Zhongwang Fu
Wuhan University
4 PUBLICATIONS **1** CITATION
SEE PROFILE

Adnan Iftekhar
Wuhan University
15 PUBLICATIONS **42** CITATIONS
SEE PROFILE

Xiaohui Cui
Wuhan University
151 PUBLICATIONS **1,735** CITATIONS
SEE PROFILE

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 1, 2020, accepted December 4, 2020, date of publication December 8, 2020,
date of current version December 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3043269

A Semantic Aware Meta-Path Model for Heterogeneous Network Representation Learning

YIPING YANG, ZHONGWANG FU, ADNAN IFTEKHAR, AND XIAOHUI CUI
Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan University, Wuhan 430000, China
Corresponding author: Xiaohui Cui (xcui@whu.edu.cn)

ABSTRACT Heterogeneous graph representation learning is to learn effective representations for nodes or (sub)graphs, which preserve node attributes and structural information. However, it is challenging to design a representation learning method for heterogeneous information networks (HINs) due to their diversity. Most of the existing HIN-oriented learning methods define a series of meta-paths. Then, they aggregate the representations learned from different meta-paths in the same hidden space. These methods do not consider semantic differences of different meta-paths, which leads to semantic confusion. And further affects the effectiveness of the learned representation. Given these issues, we introduce a Semantic Aware HIN Representation learning Network (SAHRN), which takes into account the semantics of different meta-paths. We mitigate the problem of semantic confusion by projecting nodes' features into different hidden spaces separately according to different meta-paths. To further expand the scope of aggregation and enrich the aggregated information, we also design various variants of our model by adding layer aggregation. Extensive experiments on three standard HIN datasets show that SAHRN achieves consistent improvements compared to state-of-the-art graph representation learning methods. The experiments and analyses on each component of the model show the effectiveness of the proposed method. The source code is available on https://github.com/pingpingand/SAHRN.

INDEX TERMS Data mining, graph neural networks, graph representation learning, heterogeneous networks.

I. INTRODUCTION

Most of the real-world data are organized and stored in graph structures, such as paper citation networks, knowledge graphs, social networks, and so on. It is difficult to model these non-Euclidean data directly by machine learning methods [1]. Traditional machine learning methods consider each node's features separately, limiting their abilities of processing graph-structured data. Graph representation learning aims to learn low-dimensional representations for the nodes or (sub)graphs. It can preserve the local information of the nodes as well as the global structure information. For this reason, learning an informative graph representation is of great significance [2]. It can facilitate various downstream tasks, such as node classification [3], [4], link prediction [5], [6], and recommendation system [7], [8].

There are a lot of researches on graph representation learning. They can be categorized into two classes below: **1) Random walk based methods.** Inspired by the word vector training method in word2vec [9], a series of node representation learning methods based on random walk and skip-gram have appeared [10]–[13]. These methods are intuitive and simple, but they can only preserve structural information and cannot consider node features. **2) Neural network based methods.** With the development of neural networks, some algorithms widely used in deep learning are generalized to graph-structured data. Spectral graph neural networks [14]–[16] based on spectral theory and spatial graph neural networks [17]–[19] attracted extensive attention. These methods integrate the idea of attention mechanism [20] and convolutional neural network (CNN) [21], and so on, achieved remarkable results in processing graph-structured data. However, most of these methods are designed for homogeneous graphs with only one type of node and one type of edge in a graph.

The real-world graph structures, such as knowledge graphs and social networks, often contain multiple edge and node types, which belong to heterogeneous information networks (HINs). The methods designed for homogeneous graph representation learning can not be applied to heterogeneous

The associate editor coordinating the review of this manuscript and approving it for publication was Nadeem Iqbal.

networks directly. Therefore, the research on the HIN representation learning is worthing and challenging. In this paper, we focus on the representation learning of HIN. Because of the diversity of the edges and nodes, HINs generally contain richer information than homogeneous networks. It also brings difficulties to data processing and the representation of the interaction between nodes. Here we take the HIN in Fig. 1 as an example to demonstrate this phenomenon. It includes three types of nodes (author, conference, and paper), abbreviated as A, C, and P, and two kinds of edges (write and publish). Note that we ignore the direction of the edges here. In HIN representation learning studies, meta-path [22] often represents the different semantic relations between nodes. It is the composite relationship of nodes and edges in HINs. As illustrated in Fig. 1(c), given the meta-path type Author-Paper-Author (A-P-A), there are corresponding meta-path instances in the graph, such as A1-P2-A2. The semantics of this meta-path is that two authors have published a paper together so that there is close cooperation between the authors connected by this kind of meta-path.

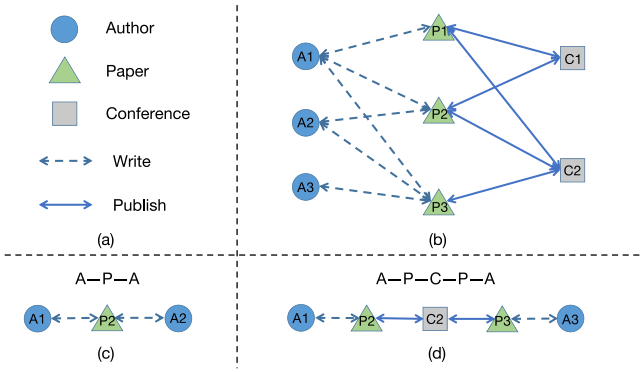

FIGURE 1. A toy example of HIN and meta-path. (a) The node and edge types. (b) An example HIN with three types of nodes and two types of edges. (c) A meta-path instance of Author-Paper-Author (APA). (d) A meta-path instance of Author-Paper-Conference-Paper-Author (APCPA).

In recent years, there have been some researches on HIN representation learning. Based on meta-path, some methods generate node sequences by meta-path guided random walk and utilize the idea of skip-gram to train node representation [23], [24]. Some methods define the measure of similarity between node pairs, and then train node embedding by combining the aggregation operation and the node sequences generated by random walk [25], [26]. Some researches combine the methods of deep neural networks, such as CNN [21], recurrent neural network (RNN) [27], and attention mechanism [20] to generate node representation and aggregate neighbors based on meta-paths [3], [28]–[30].

These methods have achieved significant performance. However, HIN representation learning still faces two main challenges:

1) For the various types of nodes and edges, we use meta-paths [22] to refine meaningful semantic representations in heterogeneous graphs, so as to make more effective use of the relationships between nodes. But not all semantics described by meta-paths are equally important for a particular task, treating them equally leads to inefficiency. Some methods based on random walk [25], [26] can not consider the different semantics and importance of meta-paths, which leads to great limitation in representation learning. For this, We consult the attention mechanism [20] and use the weighted aggregation method that similar to GAT [18] which can consider the importance of different meta-paths.

2) When performing aggregation, previous methods cannot consider each meta-path's semantics separately. They always aggregating the representation of each kind of meta-paths in the same hidden space, which leads to semantic confusion. We take the two meta-paths in Fig. 1 as a toy example, meta-path Author-Paper-Author (APA) represents the co-authoring relationship between two authors. Meta-path Author-Paper-Conference-Paper-Author (APCPA) represents that two authors published papers in the same conference. Different meta-paths will lead to entirely different semantics even the nodes connected are of the same type. Connections in different meta-paths indicate different information. If the aggregated representations are simply the weighted sum of the two meta-paths' representation, it will cause semantic confusion and lead to the problem of lacking representation ability. In addition, most neural network based methods only aggregate the information of one-hop neighbors, which may lead to insufficient aggregation of node information. However, multi-layer graph neural network has the problem of over-smoothing.

To address these problems, we propose a HIN representation learning method called Semantic Aware HIN Representation learning Network (SAHRN). Our contributions are summarized as follows:

- We highlight the importance of taking each meta-path semantically different. Through the proposed semantic aware HIN representation learning method, we can sufficiently model the different semantics of different meta-paths. It mitigates the problem of lacking node representation ability caused by the meta-path semantic confusion.
- We also propose to add a layer aggregation mechanism as variants of the model. It can integrate multi-hop neighbor relationship to further improve the effect of HIN representation learning.
- We conduct extensive experiments to illustrate the effectiveness of our proposed model. Experiments on three HIN datasets show that SAHRN achieves consistent improvements compared to state-of-the-art graph representation learning methods. The analyses and experiments on each component of the model confirm the validity of the model.

II. RELATED WORK

In this section, we review related works about HIN representation learning related to our proposed method. They are presented in two parts below.

A. GRAPH NEURAL NETWORK

With the development of neural network, the graph neural network, which extend the deep neural networks to graph data, have been intensely studied [1]. First introduced in [31] and [32], Graph Neural Network (GNN) has been combined with many different deep learning methods and achieved remarkable results. Inspired by the success of CNN [21] in the Euclidean structure, there have been many researches on generalizing CNN to graph-structured data. These studies can be categorized into two kinds of approaches: spectral based approaches [14]–[16] and spatial based approaches [17]–[19]. **1) Spectral based approaches.** To be specific, the spectral method is to realize the convolution operation on the topological graph using spectral graph theory. The idea was proposed in [33]. Until it was simplified [15], the spectral method has begun to develop vigorously. A common disadvantage of spectral methods is that they need to load the entire graph into memory to perform graph convolution, which is not efficient when processing large graphs. **2) Spatial based approaches.** The spatial method is very intuitive. In spatial based method, we update the representation of the central node by aggregating the information of neighbor nodes. Hamilton *et al.* proposed GraphSAGE [17] that learns a function to generate embedding by sampling and aggregating the features of the central node's neighbors. Velickovic *et al.* [18] applied the attention mechanism [20] to the aggregation of nodes by implicitly giving different weights to different nodes.

These methods achieve remarkable effect on homogeneous graphs, but they cannot be used on heterogeneous graphs directly.

B. GRAPH REPRESENTATION LEARNING

Graph representation learning aims to learn low-dimensional representation for individual nodes or the the entire (sub)graphs [34].

1) GRAPH REPRESENTATION LEARNING FOR HOMOGENEOUS GRAPH

There are many researches on homogeneous graph embedding learning. Inspired by word2vec [9], Perozzi *et al.* [10] first applied the idea of word embedding to graph structure. By combining the idea of random walk and skip-gram, the node vector representation is generated. Grover *et al.* proposed node2vec [11] and optimized the strategy of random walk. In Tang *et al.*'s work [12], they explicitly defined the first-order and the second-order similarity of the network to optimize the node embedding. In addition, researchers explored graph autoencoder to get the embedding by encoder-decoder structure [35], [36]. All these methods designed for homogeneous graphs have shown their potential in preserving the information of the graph structure. However, due to the heterogeneity of HINs, these methods can not be directly applied to HINs.

2) GRAPH REPRESENTATION LEARNING FOR HIN

The HIN representation learning methods can be divided into the following two categories: **1) Random walk based methods.** Different from the random walk in DeepWalk [10], Dong *et al.* [23] adopt random walk based on meta-path to handle the heterogeneity of HIN. Zhang *et al.* proposed the notion of meta-graph to enhance the flexibility and reliability of meta-path. Shi *et al.* [30] integrated embeddings of multiple meta-paths as HIN embedding, which is then used to perform prediction tasks combined with traditional matrix factorization approach. These methods are intuitive and have achieved good results, but they fail to consider the rich features of nodes, and they cannot distinguish the semantics of different meta-paths, which leads to the incomplete utilization of graph and node information. **2) Neural network based methods.** In addition to the methods based on random walk, the methods based on convolutional graph neural networks have also achieved good results recently [3], [25], [28], [29], [37], [38]. On the one hand, HetGNN [25] uses CNN [21] and bi-directional long short-term memory network (Bi-LSTM) [27] to model different types of features of nodes to generate feature vectors. On the other hand, it integrates attention mechanism [20] to aggregate different types of neighbor nodes to obtain the final vector representation. Wang *et al.* proposed HAN [28] that takes the meta-paths feature that aggregated using attention mechanism [20] as the representation of nodes. Compared with HAN, Fu *et al.* [29] adds the information of the inner nodes of meta-paths, which improves the expression ability. These methods confuse different semantic meta-paths, which impedes their representation ability. Fu *et al.* [37] considers the nodes and the relationship between nodes as a binary classification problem, which can distinguish different meta-paths' relationships, but it failed to make use of the information of nodes, which leads to limited representation ability.

In order to solve the problems in previous works, we proposed SAHRN. Our proposed method takes into account the different semantics of each meta-path and can preserve the information of each meta-path more clearly and concisely. Compared with the previous methods, we take the different semantics of each meta-paths and multi-hop relations into consideration, and improve the representation ability of the nodes.

III. PRELIMINARIES

In this section, we first introduce some terms and concepts related to heterogeneous graphs, and then make a formal problem definition of HIN representation learning.

Definition 1 (Heterogeneous Information Network (HIN)): A heterogeneous information network is defined as a directed graph $G = (V, E, \varphi, \psi)$ with node set V and edge set E. $\varphi : V \rightarrow A$ and $\psi : E \rightarrow R$ are node type mapping function and edge type mapping function, respectively. Which means each node $v \in V$ belongs to one particular node type in the node type set A, and each edge $e \in E$ belongs to one particular edge type in R, i.e., $\varphi(v) \in A$ and $\psi(e) \in R$. When $|A| +$

$|R| > 2$, the network is a HIN, otherwise, it degenerates to a homogeneous network.

Definition 2 (Meta-Path): A meta-path P is defined on the heterogeneous network G in the form $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$. It defines a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between A_1 and A_{l+1}, and can be abbreviated as $A_1A_2 \cdots A_{l+1}$. Given a meta-path P, a meta-path instance p is defined as a sequence of node types and edge types that correspond to the node types and edge types in P.

Definition 3 (Meta-Path Based Adjacency Matrix): Given the i-th meta-path P_i, $v_n \in V$ and $v_j \in V$ are connected by the meta-path, then v_j is the neighbor of v_n based on meta-path P_i. The meta-path based neighbors $N_{v_n}^{P_i}$ of v_n are defined as the set of nodes that connected with v_n by meta-path P_i. Note that $N_{v_n}^{P_i}$ includes v_n itself. Such neighborhood relation can be organized as adjacency matrix based on particular meta-path, denoted as $A^{P_i} \in \mathbb{R}^{|V_t|*|V_t|}$ where $A_{n,j}^{P_i} = 1$ if v_n and v_j are connected by meta-path P_i and $A_{n,j}^{P_i} = 0$ otherwise. And V_t is the node set of the target type.

Definition 4 (Heterogeneous Network Representation Learning): Given a HIN $G = (V, E, \varphi, \psi)$ and the feature matrix $X \in \mathbb{R}^{|V_t|*d_{A_t}}$, the HIN representation learning task is to learn low-dimensional representation $H \in \mathbb{R}^{|V_t|*d}$ which preserves both structural features and node attributes. V_t is the set of the target nodes and d_{A_t} is the initial feature dimension of them. While d is the embedding dimension with $d \ll d_{A_t}$. The representations of the nodes or the graphs can then facilitate downstream tasks, such as node classification and link prediction.

IV. PROPOSED METHOD

In this section, we present SAHRN, a semantic aware model for HIN representation learning. The main notations used in this paper are summarized in Table. 1.

TABLE 1. Main notations used in this paper.

Notations	Descriptions
G	A HIN
P_M	A set of meta-paths
P_i	The i-th meta-path
$N_v^{P_i}$	The set of neighbors of node v based on meta-path P_i
A^{P_i}	An adjacency matrix based on meta-path P_i
X	The initial feature matrix of the HIN
W^{P_i}	The projection matrix of meta-path P_i
H^{P_i}	The projected node feature matrix of meta-path P_i
$h_n^{P_i,(l)}$	The hidden representation of node n in the l-th layer
$\alpha_{n,j}^{P_i,(l)}$	The attention coefficient
H	A set containing the hidden representation of each meta-path
O^{P_i}	The predicted output based on meta-path P_i
$H^{P'}$	The final representation of the nodes

A. ARCHITECTURE OVERVIEW

The overall structure of the SAHRN model is illustrated in Fig. 2. The input of the model is the HIN $G = (V, E, \varphi, \psi)$, with the initial feature matrix denoted by $X \in \mathbb{R}^{N*d}$, where $N = |V|$ and d is the dimension of the initial feature of each target node. For each dataset, we have a manually defined a set of meta-paths, denoted as $P_M = \{P_1, P_2, \ldots, P_T\}$, where T is the number of meta-paths. Based on which we can calculate the meta-path based neighbors of each node and the meta-path based adjacency matrix set $A^P = \left\{A^{P_1}, A^{P_2}, \cdots, A^{P_T}\right\}$ of the HIN. Taking these as inputs, the SAHRN model is constructed by three components: semantic aware representation projection, semantic aware meta-path aggregation, and layer aggregation, which are introduced in Subsection IV-B, IV-C and IV-D respectively.

B. SEMANTIC AWARE REPRESENTATION PROJECTION

For different meta-paths, they have different semantics. Calculating and aggregating the representations based on meta-paths with different semantics will cause confusion. We take the meta-paths APA and APCPA in DBLP dataset as an example. APA means that the same paper connects two authors. That is, they are co-authors of an article. In comparison, APCPA means that two authors have papers published in the same conference. The meaning of the two meta-paths and the degree of the nodes based on a particular meta-path are different. Therefore, we first project the nodes' features into different hidden spaces based on the different semantics of the meta-paths to facilitate aggregation of the same kind of meta-path. Specifically, the initial feature matrix of the target node is $X \in \mathbb{R}^{N*d}$. The feature of node v_n is initialized as $h_n \in \mathbb{R}^d$. For each meta-path P_i in the meta-path set, we define projection matrix $W^{P_i} \in \mathbb{R}^{d_i*d}$ to map the feature into different semantic spaces. Where d_i is the dimension of the hidden space for meta-path P_i. The projection process is formulated as:

$$H^{P_i} = W^{P_i} \cdot X$$

where $H^{P_i} \in \mathbb{R}^{N*d_i}$ is the projected representation of the meta-path P_i in P_i's particular hidden space $\mathbb{R}^{d_i}$. And $h_n^{P_i} \in H^{P_i}$ is the mapped feature of the n-th node. For the HIN G, we denote the hidden representation set $H^P = \left\{H^{P_1}, H^{P_2}, \cdots, H^{P_T}\right\}$, T is the number of the meta-path.

C. SEMANTIC AWARE META-PATH AGGREGATION

Given a meta-path P_i, the semantic aware meta-path aggregation is to aggregate the semantic and structural information of particular meta-paths. Specifically, for the target node $v_n \in V$, P_i is the meta-path that connects it with its neighbors with particular semantics. Our purpose is to define an aggregation function so that node v_n can aggregate the neighborhood information based on meta-path P_i and learn a new representation.

For each meta-path, we aggregate the neighbors of the central node based on the meta-path. Because of HINs' heterogeneity, we can't aggregate all the inner nodes on one meta-path instance. We only take the endpoints of the meta-path as neighbors of the target nodes. Inspired by the attention mechanism [20] and its application in GAT [18], we use the method in the meta-path information aggregation stage.

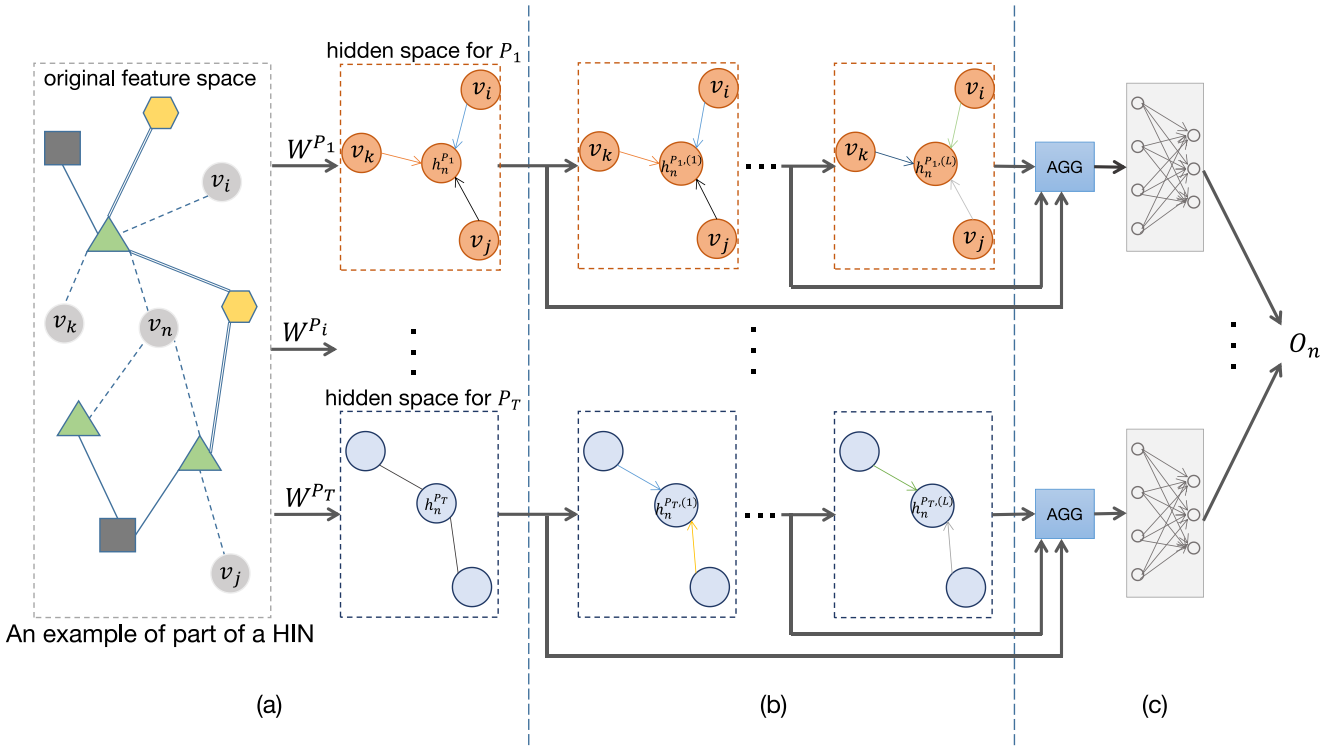

FIGURE 2. The overall architecture of SAHRN. (a) Semantic aware representation projection. For each meta-path, we project the nodes' features into different hidden spaces. (b) Semantic aware meta-path aggregation. (c) Layer Aggregation. The outputs of each layer are fed into a layer aggregation component. And then used to generate the predicted output.

By stacking the multi-layer structure, we capture the features of multi-hop neighbors. The output of each layer will be preserved for layer aggregation, which will be specified in Section IV-D. Specifically, for the n-th node in the HIN, attention mechanism [20] is used to aggregate its neighbors' features based on meta-path P_i:

$$h_n^{P_i,(l)} = \sigma\left(\sum_{j \in N_n^{P_i}} \alpha_{n,j}^{P_i,(l)} \cdot W_V \cdot h_j^{P_i,(l-1)}\right)$$

where σ is a nonlinear activation function, $N_n^{P_i}$ is the neighbors of node v_n based on meta-path P_i. $\alpha_{n,j}^{P_i,(l)}$ is the attention coefficient and represents the normalized value of the importance of node v_j. Which is calculated as follows:

$$e_{n,j}^{P_i,(l)} = LeakyRelu\left(W_Q^T \cdot \left[W_K \cdot h_n; W_K \cdot h_j\right]\right)$$

$$\alpha_{n,j}^{P_i,(l)} = \text{softmax}\left(e_{n,j}^{P_i,(l)}\right) = \frac{\exp\left(e_{n,j}^{P_i,(l)}\right)}{\sum_{j \in N_n^{P_i}} \exp\left(e_{n,j}^{P_i,(l)}\right)}$$

where [;] means concatenation operation. W_K, W_Q and W_V are trainable weight matrices. W_K and W_V are linear transformation matrices with shared parameters. $W_Q \in R^{2d_i}$, and W_K, $W_V \in R^{d_i * d}$, where d is the hidden dimension of the attention layer. To ensure the stability of training, we also use the multi-head attention mechanism [20] refer to the settings in GAT [18]. The result of the calculation by multi-head attention can be denoted as:

$$h_n^{P_i,(l)} = \Big|\Big|_{k=1}^{K} \sigma\left(\sum_{j \in N_n^{P_i}} \alpha_{n,j}^{P_i,(l),k} \cdot W_V^k \cdot h_j^{P_i,(l-1)}\right)$$

where K is the number of heads, and $||$ represents concatenation. Note that the projected representation is the input of the first layer, i.e., $h_n^{P_i,(0)} = h_n^{P_i}$. After semantic aware meta-path aggregation, we obtain the hidden representation $H = \left\{H^{(1)}, H^{(2)}, \cdots, H^{(L)}\right\}$ for each meta-path in each layer. Where L is the total number of the stacked layers, and the representation of the l-th layer is $H^{(l)} = \left\{H^{P_1,(l)}, H^{P_2,(l)}, \cdots, H^{P_T,(l)}\right\}$.

D. LAYER AGGREGATION

For the semantic aware meta-path aggregation step, the information transfer in the first layer can only consider the nodes' one-hop neighbors based on one meta-path. It may not lead to the best representation ability. If we stack multi-layer graph neural networks as the block of the model and take the final layer's representation as the output, we can consider a wider range of nodes. But it may cause the problem of gradient vanishing. This means the back-propagation through these networks will lead to over-smoothing. Eventually, the features of nodes converge to the same value [39]. That is to say, a small number of layers may lead to unstable and insufficient aggregation. At the same time, a more significant number of layers may lead to the problem of over-smoothing.

According to the argument in [40], some nodes in the graphs have few neighbors, while some nodes (hubs) are connected with many nodes at the same time. It results in the phenomenon of over-smoothing when stacking multi-layer convolution graph neural networks in the model. However, only one layer of transformation may lead to insufficient aggregation of information. In the social network from Googleplus [41], the random walk from a hub node in the center almost extends to the whole graph in four steps. In contrast, the random walk from a node with a small degree in a tree-like structure only covers a small part of all nodes [40]. Therefore, considering the different degrees and structures of nodes, the information dissemination in the network will be affected differently. In order to learn a better representation for each node, we stack multi-layer graph neural networks and keep the historical information of each layer, and finally aggregate it. In this way, we can avoid the problem of over-smoothing while aggregating multi-hop neighbor representations. Inspired by the residual connection of Resnet [42] and JK networks [40], we designed a variety of aggregation methods between layers as variants of the model.

After getting the output of each meta-path in each layer, we perform layer aggregation operation. The purpose of layer aggregation is to design an efficient aggregation method to consider each layer's information comprehensively. To avoid the insufficient aggregation of only one layer architecture and the over-smoothing of multi-layer architecture, we designed three kinds of candidate aggregation functions. Taking the meta-path P_i as an example, $H^{P_i} = \left\{H^{P_i,(1)}, H^{P_i,(2)}, \cdots, H^{P_i,(L)}\right\}$ denotes the output representation of P_i in each layer.

Average Aggregation: This function takes the element-wise average of each layer's output as the final feature of the nodes:

$$h_n^{P_i} = AVERAGE\left(\left\{h_n^{P_i,(l)}, l = 0, 1, \cdots, L\right\}\right)$$

which takes into account the features from each layer equally.

Max-Pooling: The element-wise maximize of the output of each layer is taken as the feature representation of the nodes:

$$h_n^{P_i} = MAX\left(\left\{h_n^{P_i,(l)}, l = 0, 1, \cdots, L\right\}\right)$$

which selects the most informative feature in each dimension as the aggregated feature. For the central node with a large degree and the node in the tree-like structure with a small degree, we can choose the information which contributes more to different kind of nodes.

Concatenation: It is a very direct aggregation method. It takes the concatenation of the output of each layer directly in the node feature dimension.

$$h_n^{P_i} = CONCAT\left(\left\{h_n^{P_i,(l)}, l = 0, 1, \cdots, L\right\}\right)$$

which can keep the output of each layer intact. Note that none of the three aggregation methods introduces additional parameters to learn.

After layer aggregation, the feature representation of nodes can be denoted as $H^P = \left\{H^{P_i,1}, H^{P_i,2}, \cdots, H^{P_i,T}\right\}$. For each meta-path representation, we use the fully connected layer to predict the node types distribution:

$$O^{P_i} = H^{P_i} \cdot W^{P_i}$$

where $W^{P_i} \in \mathbb{R}^{N*d_K}$. In this way, we totally get T output from the meta-paths. By adding them together, we get the final output:

$$O = \sigma\left(\sum_{i=1}^{T} y^{P_i}\right)$$

We rewrite it in the form of matrix representation for the convenient of the calculation and representation:

$$\begin{aligned} O &= \sigma\left(H^{P_1} \cdot W^{P_1} + H^{P_2} \cdot W^{P_2} + \cdots + H^{P_T} \cdot W^{P_T}\right) \\ &= \sigma\left(\left[H^{P_1}; H^{P_2}; \cdots; H^{P_T}\right] \cdot \left[{W^{P_1}}^T; {W^{P_2}}^T; \cdots; {W^{P_T}}^T\right]\right) \\ &= \sigma\left(H^{P'} \cdot W^{P'}\right) \end{aligned}$$

where σ is a nonlinear activation function. $W^{P'} \in \mathbb{R}^{dKT*d_{out}}$, d_{out} is the number of categories in the classification task, d is the hidden dimension of the attention layer. And the final representation learned in our model is denoted as $H^{P'} \in \mathbb{R}^{N*dKT}$.

E. TRAINING

After the projection and aggregation as mentioned above, we get the final node feature representation, which can be then used for various downstream tasks.

Here, we take semi-supervised node classification as the downstream task to train the model. We use part of the labeled nodes as the training set. Then we calculate the classification loss of the final output O and the actual node label using the cross entropy loss, and minimize it through back-propagation and gradient descent. The loss function can be formulated as:

$$L = -\sum_{i \in V_t} y_i \cdot \log O_i$$

where V_t is the set of labeled nodes in the training set. O_i and y_i are the predicted and the ground truth class label for node i, respectively.

V. EXPERIMENTS

In this section, we conduct extensive experiments and analyses to demonstrate the effectiveness of SAHRN on HIN representation learning. We first introduce three publicly available real-world HIN datasets and the baseline models for representation learning. Then we evaluate our proposed models and the baseline models in two downstream tasks. Finally, we perform a series of detailed experiments on each component of the model to analyze the effectiveness of the proposed method.

A. DATASETS

We adopt three widely used real-world HIN datasets to evaluate the performance of our proposed method and the baseline

TABLE 2. Statistics of datasets.

Dataset	Number of Nodes	Number of Edges	Feature Dimension	Training Set	Validation Set	Test Set	Meta-path Set
IMDB	12890	19120	1232	300	300	2687	MAM MDM
DBLP	27194	122393	334	800	400	2857	APA APCPA APTPA
ACM	8916	12769	1830	600	300	2125	PAP PSP

models. The datasets are provided in HAN [28] and are accessible from the original paper. We conduct experiments on node classification and node clustering in these datasets. Simple statistics of these datasets are summarized in Table. 2. Among them, the choice of meta-paths is based on domain knowledge and experimental results.

IMDB:[1] It is a collected subset from the IMDB which contains nodes of movies, directors, and actors, denoted as M, D, and A. Movies are the target nodes that can be categorized into three classes based on their genre: action, comedy, and drama. We utilize the plots keywords of the movies represented by bag-of-words as features. Meta-path set {'MAM', 'MDM'} is employed to perform the experiments.

DBLP:[2] It is an academic paper dataset. We use the extracted subset of DBLP which contains nodes of papers, authors, terms and conferences, denoted as P, A, T, and C respectively. This dataset consists of 20 conferences from four different research areas: database, data mining, machine learning, and information retrieval. The author's nodes are the target nodes. Each author is labeled based on their research field according to the conferences they submitted paper in. And we use the elements of a bag-of-words represented of keywords. Meta-path set {'APA', 'APCPA', 'APTPA'} is employed to perform the experiments.

ACM:[3] It is another academic paper dataset, which contains nodes of papers, authors, and subjects, denoted as P, A, and S. The paper's nodes are the target nodes. They are categorized into three classes: database, wireless communication, and data mining. The initial features of the paper nodes are represented as bag-of-words of keywords. Meta-path set {'PAP', 'PSP'} is employed to perform the experiments.

B. COMPARED METHODS

We compared our proposed SAHRN with some state-of-the-art baselines in graph representation learning. These models include random walk based models as well as GNN based models.

DeepWalk [10]: A random walk based method. It is designed for homogeneous graphs. We perform DeepWalk in each meta-path based adjacency matrix. And finally report the best performance.

GCN [15]: A GNN based semi-supervised method for homogeneous graphs. We test GCN on each meta-path based adjacency matrix and report the best performance.

R-GCNs [38]: A GNN based method. The weights of different types of edges can be considered.

HetGNN [25]: A GNN based method for heterogeneous graphs, which can comprehensively consider node heterogeneous content information and neighbor structure to generate node embeddings.

GAT [18]: A GNN based method for homogeneous graphs. It utilizes the attention mechanism [20] to perform spatial domain convolution operation. We also test GAT on each meta-path based adjacency matrix and report the best performance.

HAN [28]: A GNN based method for heterogeneous graphs. It learns node representations from each meta-path. By aggregating these representations using attention mechanism [20], it generates representation vectors for each node.

SAHRN (Our Proposed Method): A graph representation learning method designed for HINs. It can distinguish the different semantics of each meta-path.

SAHRNavg: A variant of our proposed model. Layer aggregation is added to the model to fuse multi-hop neighbors' information. It uses average aggregation function.

SAHRNmxp: A variant of our proposed model with added Layer aggregation. It uses max-pooling aggregation function.

SAHRNcat: A variant of our proposed model with added Layer aggregation. It uses concatenation aggregation function.

C. HYPER-PARAMETER SETTINGS

In the stage of model training, the training, test, and validation set are pre-segmented. In order to ensure the comparability of the models' effects, we use the same data segmentation in each model. The split ratio and the meta-paths we use are shown in Table. 2.

For our proposed method SAHRN, the model is implemented with Tensorflow, a python-based framework. Detailed implementation can be found in the source code we provide. We randomly initialize parameters. For optimization, we set the learning rate of 0.005, the l2 regularization weight of 0.001, and the optimize function of Adam [43]. The number of attention head of 8. We use early stopping and set the number of epochs to 100, the patience of 5. The nonlinearity function is set to ELU [44]. The set of parameters in the attention aggregation is the same as GAT [18]. To make the model easy to tune, we set the same hidden projection

[1]https://www.imdb.com
[2]https://dblp.uni-trier.de
[3]http://dl.acm.org

TABLE 3. Node classification results.

Dataset	Metric (%)	Training	DeepWalk	GCN	R-GCN	Het-GNN	GAT	HAN	SAHRN-avg	SAHRN-mxp	SAHRN-cat	SAHRN
IMDB	Micro-F1	20%	41.41	55.40	49.54	56.80	56.91	59.17	59.63	59.22	57.81	**60.19**
		40%	41.98	55.34	51.20	56.09	56.96	59.77	59.75	60.45	59.94	**61.18**
		60%	42.44	55.60	53.93	56.79	56.93	60.28	61.19	61.13	59.46	**61.99**
		80%	43.46	55.33	55.95	56.82	56.77	59.59	60.99	61.65	**62.08**	61.99
	Macro-F1	20%	36.87	47.65	48.31	47.81	48.71	52.54	53.93	55.00	**55.39**	55.34
		40%	37.21	47.19	50.06	48.19	49.40	53.91	54.50	56.30	**56.81**	56.62
		60%	37.21	48.64	52.63	48.13	48.91	54.67	56.02	57.22	**57.39**	57.30
		80%	37.94	47.98	54.54	48.69	49.31	53.81	55.81	**57.93**	57.90	57.28
DBLP	Micro-F1	20%	91.29	92.58	91.66	92.66	92.00	93.07	93.59	93.43	93.25	**93.65**
		40%	91.72	92.43	91.82	92.76	92.29	93.20	93.92	93.87	93.63	**93.97**
		60%	92.64	92.70	92.16	92.73	92.41	93.23	94.18	93.94	93.74	**94.25**
		80%	92.33	92.66	92.55	93.13	92.57	93.48	93.74	94.04	93.90	**94.86**
	Macro-F1	20%	90.29	91.64	90.53	91.65	90.78	92.19	92.83	92.66	92.40	**92.86**
		40%	90.70	91.53	90.72	91.71	91.15	92.29	93.16	93.08	92.88	**93.19**
		60%	91.74	91.84	91.20	91.65	91.36	92.35	93.44	93.18	93.08	**93.55**
		80%	91.37	91.74	91.65	92.09	91.43	92.61	93.28	93.23	93.11	**94.25**
ACM	Micro-F1	20%	68.15	87.66	86.98	87.72	87.86	89.34	90.24	89.95	**90.57**	90.15
		40%	70.89	88.12	88.76	87.96	88.16	90.19	90.45	89.84	**90.77**	90.47
		60%	71.04	88.35	89.89	88.26	87.98	90.65	**91.05**	90.29	90.15	90.52
		80%	71.88	88.45	90.12	88.68	88.75	89.93	90.78	90.31	**91.22**	90.73
	Macro-F1	20%	68.04	87.62	87.13	87.71	87.80	89.44	90.36	90.06	**90.67**	90.16
		40%	70.77	88.11	88.90	87.96	88.12	90.27	90.55	89.92	**90.85**	90.47
		60%	70.95	88.33	90.08	88.29	87.90	90.74	**91.13**	90.36	90.29	90.56
		80%	71.78	88.41	90.28	88.76	88.60	90.11	90.90	90.38	**91.30**	90.77

TABLE 4. Node clustering results.

Dataset	Metric	DeepWalk	GCN	R-GCN	Het-GNN	GAT	HAN	SAHRN
IMDB	NMI	0.0087	0.0737	0.0081	0.0934	0.0799	0.1081	**0.1125**
	ARI	0.0073	0.0723	0.0100	0.0921	0.0725	0.0968	**0.0970**
DBLP	NMI	0.7322	0.6981	0.1834	0.7576	0.6951	0.7681	**0.7883**
	ARI	0.7745	0.7513	0.1513	0.8190	0.7465	0.8308	**0.8446**
ACM	NMI	0.2180	0.5448	0.3901	0.5930	0.5796	0.6008	**0.6201**
	ARI	0.2256	0.5780	0.3349	0.5889	0.6137	0.6142	**0.6236**

dimension for each meta-path in one dataset. In DBLP, ACM and IMDB, the dimension are set to 500. The number of layers of these three datasets are set to 2.

For baseline models, we train the GAT and GCN for 100 epochs with the early stopping patience set to 5. Except for special instructions, we keep the settings as in the original paper. For the methods designed for homogeneous networks, we perform experiments on each meta-path based adjacency matrix, and report the best performance.

D. PERFORMANCE COMPARISON

In this subsection, we evaluate the effectiveness of the node representation learned by our proposed models with the baseline models. The node representation is used for node classification and clustering tasks respectively.

Fig. 3 shows the experimental results on three datasets. Our original model achieves better results on the three datasets. However, the accuracy of node classification of the variant model without semantic aware projection has shown a significant decrease. This confirms the necessity of semantic aware projection in our proposed model. Distinguishing the semantics of different meta-paths can significantly improve the representation effect of node embedding.

1) NODE CLASSIFICATION

Here we employ support vector machine (SVM) classifier to perform node classification. We set varying training proportions for each dataset, 20%, 40%, 60%, and 80%. For the stability of the results, we repeat the experiment ten times on each dataset and report the average Macro-F1 and Micro-F1. The experiment results are shown in Table. 3.

As shown in Table. 3, SAHRN and its variants achieve the best performance. Compared with the baseline models, the experimental results of our proposed method on the three datasets show significant improvement.

For the baseline methods, the GCN, GAT, R-GCNs and HetGNN that based on GNNs perform better than the method based on random walk. This is because GCN [15] and GAT [18] integrated the idea of CNN [21] and attention mechanism respectively, which can better aggregate the information of nodes. What's more, they can consider the features of nodes and graph-structured information simultaneously. In comparison, DeepWalk [10] can only consider the graph structural information. Therefore, GNN based models show better performance. Compared with the models designed for homogeneous graphs, HAN [28] performs better in the three datasets. It can integrate information from different meta-paths. Through the attention mechanism, the information of multiple meta-paths is fused in the same hidden space. The difference in importance of each meta-path can also be considered.

Our proposed model can take the semantics of different meta-paths into account and calculate them separately in different hidden spaces, so it achieves the best results. Note

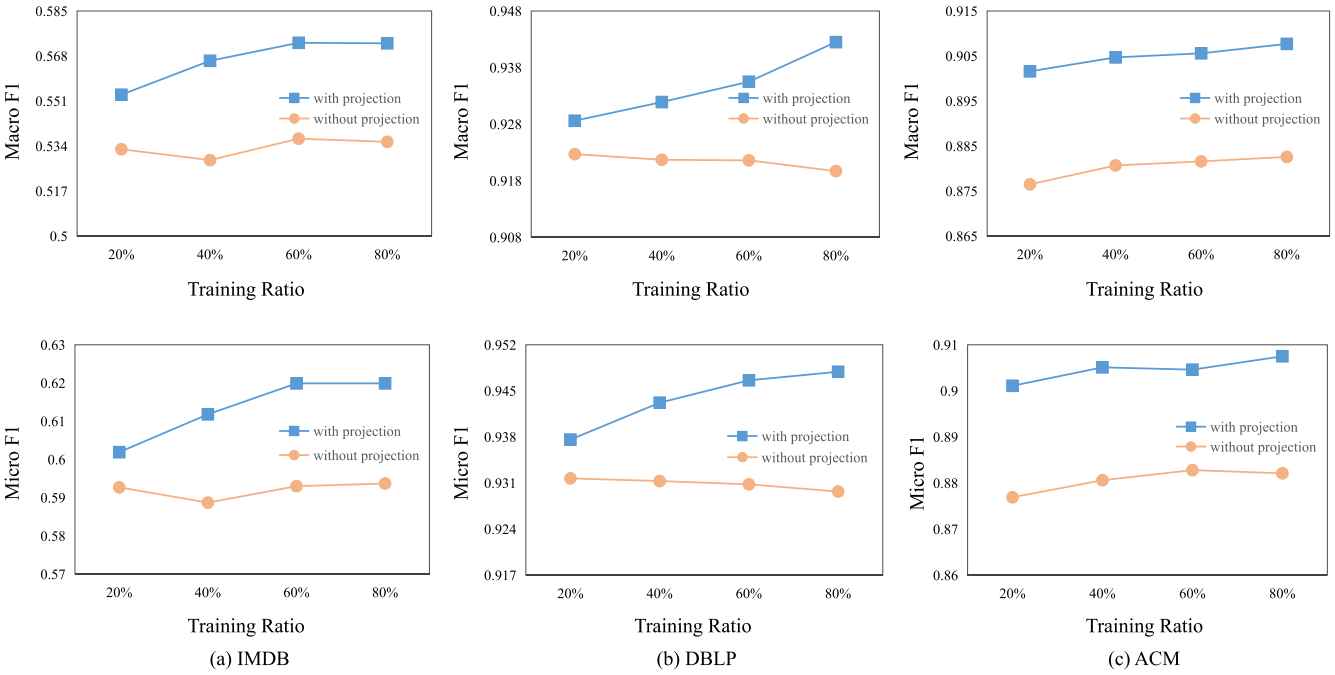

FIGURE 3. Performance comparison of the original model and its variant without semantic aware projection.

that in DBLP, the SAHRN without layer aggregation achieves the best result. While in ACM and IMDB, the variants of SAHRN that add layer aggregation achieve better results in most cases. This may be because of the difference in the graph structure. We will leave this phenomenon to Subsection V-E for further discussion.

2) NODE CLUSTERING

We employ the K-Means algorithm to compare the performance of each model in node clustering. The number of clusters K is set to the categories of the target node in each dataset. To measure the clustering effect, we use normalized mutual information (NMI) and adjusted rand index (ARI) as the evaluation methods. For the stability of the results, we repeat the experiment ten times on each dataset and report the average NMI and ARI. The results are shown in Table. 4. Here, in order to facilitate comparison, we only use the basic model for comparison and analysis.

As shown in Table. 4, we can see that SAHRN consistently performs better than the baseline models. The node representation learned by our proposed method can be effectively applied to node clustering tasks. The performances of the model designed for homogeneous graphs are significantly worse than that of the HIN representation learning model. That's because they cannot take into account the rich semantics of meta-paths. On the other hand, HAN aggregates the representation of each meta-path, and confuses semantics. Our proposed method can distinguish meta-path semantics by projecting the node features into different hidden spaces based on the meta-paths. Therefore, it gets a better node representation and outperforms the baseline model.

E. DETAILED DISCUSSION OF SAHRN

In this subsection, we perform detailed analyses and experiments to confirm the effectiveness of each component of our proposed method.

1) EFFECTIVENESS OF SEMANTIC AWARE PROJECTION OPERATIONS

In our basic model, we use semantic aware projection for the nodes based on each meta-path. By projecting node features into different hidden spaces, we can avoid semantic confusion. To validate the effectiveness of this component, we design a variant of the model. Here SAHRN is our basic model that contains the projection component. SAHRN-project is the variant that removes the projection operation from the original model. Like the original model, we perform node aggregation based on each meta-path using the attention mechanism in this variant. Besides, we take the average of the embeddings based on each meta-path to obtain the final representation of the node. We still use node classification to measure the effect of the models. The task settings are the same as before. The experimental results are shown in Fig. 3.

Fig. 3 shows the experimental results on three datasets. Our original model achieves better results on the three datasets. However, the accuracy of node classification of the variant model without semantic aware projection has shown a significant decrease. This confirms the necessity of semantic aware projection in our proposed model. Distinguishing the semantics of different meta-paths can significantly improve the representation effect of node embedding.

2) EFFECTIVENESS OF LAYER AGGREGATION

In the graphs, the neighbor structure of each node is different. So as the degree of the nodes. There will also be differences

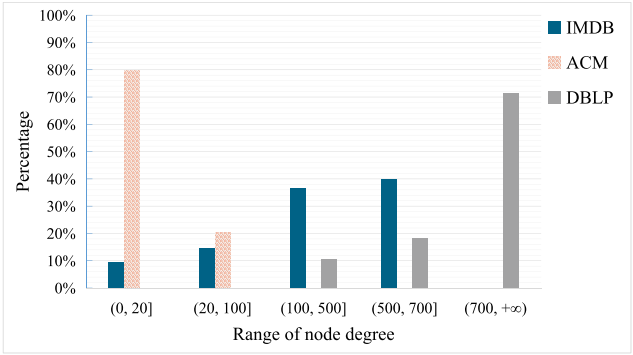

FIGURE 4. **The degree of nodes in each dataset.**

in the dissemination of information between nodes. Some nodes (hubs) are connected with a great amount of nodes at the same time. If we stack multi-layer GNNs, it will result in over-smoothing. While in some graphs, the nodes have few neighbors, only one layer of transformation may lead to insufficient aggregation of information.

Based on our node classification results in V-D, in ACM and IMDB, the variants of SAHRN that add layer aggregation achieve better results in most cases. While in DBLP, the variants do not work well. We suspect that the different properties of the datasets cause this. Therefore, we conduct experiments to separately calculate the degree distribution of the nodes in these three datasets. We use meta-path based adjacency matrix in the previous experiment. Moreover, we calculate the average degree of the nodes based on different meta-paths in each dataset. The statistical results are shown in Fig. 4.

In DBLP, most nodes have a degree of more than 500, which are mostly hub nodes with many neighbors, so information spreads faster in DBLP. While the nodes in ACM and IMDB generally have a smaller degree. They are more likely to be in a tree-like structure. Experiments in [40] show that random walk starting from this kind of node within a specified number of steps cover much fewer nodes in the graph than that from the hub nodes. It also confirms our conjecture. This difference in the dataset structure leads to the different effects of the model and its variants. According to the characteristics of the dataset, stacking multi-layer graph neural networks and selecting the appropriate number of layers will improve the effect of node representation learning.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a graph representation learning model, SAHRN. It learns node representations for heterogeneous information networks. We design a semantic aware projection operation for SAHRN. For different meta-paths, the nodes are projected into different hidden spaces to avoid semantic confusion of meta-paths. Considering the difference of information propagation in each dataset, we also design a series of variants of our model. By stacking multiple layers of GNNs and preserving information from each layer, we can avoid the problem of insufficient information dissemination or over-smoothing caused by multi-layer structure. Extensive experiments on three datasets show that our proposed model and its variants achieve better performance than state-of-the-art baseline models. Our analyses and experiments on each component of the model confirm the validity of the proposed model.

In the future, we plan to investigate how to utilize the information inside the meta-paths more comprehensively. It may make it possible for us to preserve the semantics of the meta-paths better.

REFERENCES

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 24, 2020, doi: 10.1109/TNNLS.2020.2978386.

[2] Y. Dong, Z. Hu, K. Wang, Y. Sun, and J. Tang, "Heterogeneous network representation learning," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Yokohama, Japan, Jul. 2020, pp. 4861–4867.

[3] H. Hong, H. Guo, Y. Lin, X. Yang, Z. Li, and J. Ye, "An attention-based graph neural network for heterogeneous structural learning," in *Proc. AAAI Conf. Artif. Intell.*, New York, NY, USA, 2020, pp. 4132–4139.

[4] Y. Ren, B. Liu, C. Huang, P. Dai, L. Bo, and J. Zhang, "Heterogeneous deep graph infomax," 2020, *arXiv:1911.08538*. [Online]. Available: https://arxiv.org/abs/1911.08538

[5] C. Zhang, A. Swami, and N. V. Chawla, "SHNE: Representation learning for semantic-associated heterogeneous networks," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Melbourne, VIC, Australia, Jan. 2019, pp. 690–698.

[6] Z. Li, Z. Liu, J. Huang, G. Tang, Y. Duan, Z. Zhang, and Y. Yang, "MV-GCN: Multi-view graph convolutional networks for link prediction," *IEEE Access*, vol. 7, pp. 176317–176328, 2019.

[7] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Paris, France, Jul. 2019, pp. 165–174.

[8] T. Wang, D. Shi, Z. Wang, S. Xu, and H. Xu, "MRP2Rec: Exploring multiple-step relation path semantics for knowledge graph-based recommendations," *IEEE Access*, vol. 8, pp. 134817–134825, 2020.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Workshop Int. Conf. Learn. Represent. (ICLR)*, Scottsdale, Arizona, 2013, pp. 1–12.

[10] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York City, NY, USA, 2014, pp. 701–710.

[11] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 855–864.

[12] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, May 2015, pp. 1067–1077.

[13] J. Tang, M. Qu, and Q. Mei, "PTE: Predictive text embedding through large-scale heterogeneous text networks," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Sydney, NSW, Australia, 2015, pp. 1165–1174.

[14] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst. (NPIS)*, Barcelona, Spain, 2016, pp. 3837–3845.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, 2017, pp. 1–13.

[16] F. Wu, T. Zhang, A. H. D. Souza, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *Proc. Int. Conf. Mach. Learn (ICML)*, Long Beach, CA, USA, 2019, pp. 6861–6871.

[17] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Proces. Syst (NIPS)*, Long Beach, CA, USA, 2017, pp. 1024–1034.

[18] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, 2018, pp. 1–12.

[19] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, 2018, pp. 1–15.
[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.
[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Proces. Syst. (NIPS)*. Lake Tahoe, NV, USA: Harrahs and Harveys, 2012, pp. 1097–1105.
[22] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," in *Proc. VLDB Endownment*, Seattle, WA, USA, 2011, pp. 992–1003.
[23] Y. Dong, N. V. Chawla, and A. Swami, "Metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Halifax, NS, Canada, Aug. 2017, pp. 135–144.
[24] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Metagraph2vec: Complex semantic path augmented heterogeneous network embedding," in *Proc. PAKDD*, Melbourne, VIC, Australia, 2018, pp. 196–208.
[25] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Anchorage, AK, USA, 2019, pp. 793–803.
[26] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, and J. Tang, "Representation learning for attributed multiplex heterogeneous network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Anchorage, AK, USA, Jul. 2019, pp. 1358–1368.
[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[28] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *Proc. World Wide Web Conf.*, San Francisco, CA, USA, May 2019, pp. 2022–2032.
[29] X. Fu, J. Zhang, Z. Meng, and I. King, "MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proc. Web Conf.*, Taipei, Taiwan, Apr. 2020, pp. 2331–2341.
[30] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.
[31] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Montreal, QC, Canada, Jul./Aug. 2005, pp. 729–734.
[32] F. Scarselli, M. Gori, A. Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
[33] J. B. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, Banff, AB, Canada, 2014, pp. 3837–3845.
[34] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Eng. Bull.*, vol. 40, no. 3, pp. 52–74, 2017.
[35] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *Proc.27th Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, Jul. 2018, pp. 1–7.
[36] K. Tu, P. Cui, X. Wang, P. S. Yu, and W. Zhu, "Deep recursive network embedding with regular equivalence," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., Jul. 2018, pp. 2357–2366.
[37] T.-Y. Fu, W.-C. Lee, and Z. Lei, "HIN2 Vec: Explore meta-paths in heterogeneous information networks for representation learning," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Singapore, Nov. 2017, pp. 1797–1806.
[38] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. V. D. Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proc. Eur. Semantic Web Conf.* Cham, Switzerland: Springer, 2018, pp. 593–607.
[39] Q. Li, Z. Han, and X. M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. Workshop AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, 2018, pp. 3538–3545.
[40] K. Xu, C. Li, Y. Tian, T. Sonobe, K. I. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proc. Int. Conf. Int. Conf. Mach. Learn (ICML)*, Stockholm, Sweden, 2018, pp. 5453–5462.
[41] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Proc. Adv. Neural Inf. Proces. Syst (NIPS)*. Lake Tahoe, NV, USA: Harrahs and Harveys, 2012, pp. 539–547.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015, pp. 1–15.
[44] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUS)," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, 2016, pp. 1–14.

YIPING YANG received the bachelor's degree in engineering management from the Beijing University of Posts and Telecommunications, Beijing, China, in 2019. She is currently pursuing the master's degree with Wuhan University. Her research interests include nature language processing and data mining.

ZHONGWANG FU received the master's degree from the School of Computer Science and Information Engineering, Hubei University, in 2018. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Wuhan University. His research interests include information security, social computing, and big data.

ADNAN IFTEKHAR received the M.Sc. degree in computer science from COMSATS University Islamabad, in 2016. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Wuhan University. He is currently a Lecturer with the Department of Computer Science, Faculty of Information and Technology. His research interests include applications of distributed ledger technology, the Internet of Things, and the blockchain in the food safety and food security area.

XIAOHUI CUI received the Ph.D. degree in computer science and engineering from Louisville University, in 2004. He was a Visiting Professor with Louisville University, the Head of Food Safety BlockChain Alliance, the Vice President of Demonstrational Software School Alliance, the member of High-Performance Computing Special Project of National Key Research and Development Program, the member of Advisory Committee of Software Engineering Teaching Program, the member of CSC, and the member of Computing Service with CCF. He was also the Dean of the School of International Software, Wuhan University, and a Researcher of Computational Data Analysis Department with the Oak Ridge National Laboratory. He is currently a second grade Professor with the School of Cyber Science and Engineering, Wuhan University.