IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 17, 2019, accepted September 8, 2019, date of publication September 18, 2019,
date of current version September 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2942221

DHNE: Network Representation Learning Method for Dynamic Heterogeneous Networks

YING YIN[1], LI-XIN JI[1], JIAN-PENG ZHANG[1], AND YU-LONG PEI[2]
[1]Information Engineering University, Zhengzhou 450003, China
[2]Eindhoven University of Technology, Eindhoven, The Netherlands

Corresponding author: Ying Yin (15883880517@163.com)

This work was supported in part by the National Nature Science Foundation of China for the Innovation Research Group under Grant 61521003.

ABSTRACT Analyzing the rich information behind heterogeneous networks through network representation learning methods is signifcant for many application tasks such as link prediction, node classifcation and similarity research. As the networks evolve over times, the interactions among the nodes in networks make heterogeneous networks exhibit dynamic characteristics. However, almost all the existing heterogeneous network representation learning methods focus on static networks which ignore dynamic characteristics. In this paper, we propose a novel approach DHNE to learn the representations of nodes in dynamic heterogeneous networks. The key idea of our approach is to construct comprehensive historical-current networks based on subgraphs of snapshots in time step to capture both the historical and current information in the dynamic heterogeneous network. And then under the guidance of meta paths, DHNE performs random walks on the constructed historical-current graphs to capture semantic information. After getting the node sequences through random walks, we propose the dynamic heterogeneous skip-gram model to learn the embeddings. Experiments on large-scale real-world networks demonstrate that the embeddings learned by the proposed DHNE model achieve better performances than state-of-the-art methods in various downstream tasks including node classifcation and visualization.

INDEX TERMS Dynamic heterogeneous networks, network representation learning, random walk, skip-gram model.

I. INTRODUCTION

Social communication systems, academic information systems, and biomedical systems are very common in our real life. How to analyze characteristics of these systems is of vital importance to solve practical application problems. With the rise of network science, these systems can be modeled into the form of complex networks. Research on complex networks can help us analyze the characteristics of these systems effectively. Then, how to model these systems into complex networks needs to be taken into account first. A simple way is to model the entities in the system as nodes and model the relationships between entities as edges, and all the nodes and edges are treated as a single type. In this way, systems can be modeled as homogeneous information networks. Although there are many researches on homogeneous networks, they can't capture the rich information contained in some real systems completely. Real-world applications are often oriented to systems with multiple types of entities and complex link relationships. Treating all the nodes and edges in the systems as the same type will lose lots of important information such as semantic information in academic information systems.

By structuring data objects and their interactions in the systems into multiple types of nodes and edges, complex systems can be expressed in the form of heterogeneous information networks which can preserve rich information in the original systems. Mining the rich information contained in heterogeneous network is important for us to do the network application tasks such as node classification [1], similarity research [2], and link prediction [3]. In recent years, machine learning methods have been widely used to deal with network application tasks. In order to represent networked data in a reasonable form as the input of machine learning methods to improve the performance of these algorithms, network representation learning is gradually emerging.

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

Network representation learning maps nodes in a network to low-dimensional spaces in order to form low-dimensional dense vectors that can be used as the input of machine learning models to do the downstream tasks. It serves as a bridge connecting the networked data analysis, traditional machine learning, and data mining algorithms.

Most of the existing network representation learning methods focus on static heterogeneous networks which only consider the static structural properties at a certain moment [4], [5]. However, as the network evolves over times, the interactions among the nodes in networks make networks exhibit dynamic characteristics. For example, in the academic information network which is a typical heterogeneous network, as the author's research direction changes, the co-authorship will also change. But if we just analyze academic networks in the static condition, we can never get these dynamic information. Therefore, the representation learning of complex heterogeneous networks with dynamic properties can better mine the rich information contained in networks.

To capture the dynamic information in the heterogeneous network, we proposed the dynamic heterogeneous network representation learning method to learn heterogeneous network embeddings from a dynamic perspective, namely DHNE(Dynamic Heterogeneous Network Embedding). The general idea of our approach is to construct historical-current graphs to preserve both the historical information at the previous time steps and the current information in the heterogeneous network. Based on the historical-current graphs, we perform random walks under the guidance of meta paths which contain different semantic information and then we propose the dynamic heterogeneous skip-gram model to learn the representations of nodes in the dynamic heterogeneous network.

It is worthwhile to highlight our contributions as follows:

- To the best of our knowledge, this is the first work that learns the embeddings of dynamic heterogeneous networks to preserve both the current and historical characteristics of nodes over a given time step.
- Since we combine multiple types of information including current information, historical information and semantic information to learn node embeddings, nodes can be reasonably represented and can be applied to mine dynamic characteristics of the network.
- We construct experiments over two real-world datasets, and the experimental results demonstrated that the embeddings learned from the proposed DHNE model can achieve better performances than the state-of-the-art methods in downstream tasks, such as node classification and visualization.

The rest of the paper is organized as follows. Section II provides an overview of related work. We present the problem statement in Section III. Section IV explains the technical details of *DHNE*. In Section V we then discuss our experimental study. Finally, in Section VI we draw conclusions and outline directions for future work.

II. RELATED WORK

In the era of big data, the research and analysis of networked data have been widely concerned in various fields. Complex and huge networked data is often difficult to process. It is of great significance to represent networked data as an efficient and reasonable form to solve practical application problems. Network representation learning is a distributed learning method that maps nodes in a network to low-dimensional spaces in order to form vectors with certain reasoning ability. The vectors obtained by network representation learning can be conveniently used as the input to machine learning models to solve downstream tasks.

Traditional network representation learning focus on homogeneous information networks which consist of singular type of nodes and relationships. For instance, Perozzi et al. applied "shallow" neural network to homogeneous network representation learning and proposed the classic DeepWalk algorithm [6] which is based on the Word2vec [7] model in natural language processing. Grover et al. proposed node2vec algorithm [8] to improve the random walk process of DeepWalk which preserves the homogeneity and isomorphism of network by combining breadth-first search and depth-first search. With the development of deep learning, Wang *et al.* proposed the SDNE algorithm [9] to apply the deep neural network to homogeneous network representation learning. The semi-supervised deep learning model preserves the local and global information of network through the first-order similarity module and the second-order similarity module, respectively. The GraphGAN model [10] proposed by Wang et al. and the ANE model [11] proposed by Dai et al. applied the GAN [12] model into network representation learning, which greatly improved the robustness of homogeneous network embedding. For homogeneous information networks, some of the existing network representation learning methods have taken the dynamic characteristics into account which are mostly derived from the static homogeneous network representation learning model. Inspired by the static network representation learning method based on matrix decomposition, Li et al. proposed DANE algorithm [13] to leverage the variation of the adjacency matrix as well as the attribute matrix of the network snapshots to update the current node embeddings based on the node embeddings at the previous time. Cui et al. proposed the DHPE algorithm [14] to update the node embeddings dynamically based on generalized singular value decomposition (GSVD) and matrix perturbation theory while preserving the high-order proximity. In addition to methods based on matrix decomposition, there are some approaches extending the classic models to learn the dynamic network representations. DNE algorithm [15] proposed by Du et al. extends the skip-gram model to dynamic homogeneous network representation learning which learns the embeddings of new vertices by only updating the embeddings of partial vertices to improve the efficiency of the algorithm. Nguyen et al. proposed CTDNE [16] to apply the skip-gram model to temporal network embedding. This approach presents dynamic networks in the form of networks with multiple timestamps on each edge which indicate when the interactions occurred. On this basis, CTDNE specifies that each random walk must conform to the chronological order in which the interactions took place in order to capture the temporal information through the random walks.

We can see from the above homogeneous network representation learning methods and the review of predecessors [17]–[19], homogeneous network representation learning has gradually matured. However, structuring complex systems in the form of homogeneous information networks that treat all the nodes and connected edges as the same type ignores the rich semantic information in the network. Representing the real-world systems in the form of heterogeneous information networks is more realistic and practical. In order to mine the rich information contained in the heterogeneous information networks, the heterogeneous network representation learning methods have come out in recent years. Inspired by the node2vec algorithm, Dong et al. proposed the metapath2vec algorithm [4] which acquires the "context" of nodes by performing random walks in the heterogeneous information network under the guidance of meta path [2] which is an important characteristic of heterogeneous information networks. Tang et al. proposed PTE [5] to extend LINE [20] to heterogeneous text network representation learning which leverages both labeled and unlabeled data to learn the embeddings in a semi-supervised manner. PTE can learn the characteristics of labeled data while preserving the advantages of unsupervised embedding. Chang et al. designed a deep embedding algorithm [21] to map different types of nodes in the heterogeneous network into the same low dimensional vector space. The proposed deep architecture model preserves both the local and global linkage structures of the heterogeneous network which makes it more powerful to capture information in the heterogeneous network. There are also some heterogeneous network representation learning methods combined with specific application tasks. The Heterecom [22] model proposed by Shi et al. combines the embeddings with specific downstream task recommendation. The approach extracts the same type of node sequences from the heterogeneous information network based on different meta paths to b24process, which is equivalent to extracting multiple homogeneous information networks from the heterogeneous information network. Chen et al. proposed a task-guided heterogeneous network embedding method [23] for author identification which is posed as author ranking problem in heterogeneous networks. This approach extends the existing unsupervised network embedding model to incorporate meta paths under the guidance of the author identification task. The embeddings are learned by both task-specific and network-general objectives.

However, as the heterogeneous networks evolve over time, all the aforementioned network representation learning methods are no longer applicable. In order to analyze the heterogeneous network in the dynamic condition, we first propose DHNE to combine both the historical and current information in the dynamic heterogeneous network to learn the unified representations for nodes.

III. PROBLEM STATEMENT

In this section, we formulate the problem of dynamic heterogeneous information network embedding. First, we give necessary definitions used throughout this paper as follows:

Definition 1 (Dynamic Heterogeneous Information Network): A dynamic heterogeneous information network within time T can be defined as a collection $G = \{G^1, G^2, \dots, G^T\}$ containing a series of time snapshots. The snapshot at time t is a heterogeneous information network that can be denoted as $G^t = (V_t, E_t, f, \gamma)$, where V_t and E_t denote the set of nodes and edges at time t respectively, $f : V_t \rightarrow A$ is the node type mapping function, where each node $v \in V_t$ corresponds to a specific type in A. $\gamma : E_t \rightarrow R$ is the edge type mapping function, where each link $e \in E_t$ corresponds to a specific type in R. In the heterogeneous information network, $|A| + |R| > 2$.

Definition 2 (Network Representation Learning): Given a network $G = (V, E)$, where E represents a collection of nodes, E represents a collection of edges. Network representation learning maps nodes $v \in V$ in the original space to low-dimensional spaces in the form of low-dimensional dense vectors $r_v \in R^k$ (where $k \ll |V|$) through a mapping function, such that the structural and semantic relations in the original network can be preserved in R^k.

Definition 3 (Dynamic Heterogeneous Information Network Representation Learning): Given a dynamic heterogeneous network $G = \{G^1, G^2, \dots, G^T\}$, we map nodes in the snapshots to the low-dimensional space so that nodes can be represented as vectors and the same node at different times can be represented as different vectors, so that the temporal and structural information can be preserved in the low-dimensional vector space.

Definition 4 (Meta Path): In the heterogeneous information networks, node v_i and v_j may be connected via multiple paths. In Figure 3, authors A_2 and A_4 are connected by $A_2 \xrightarrow{R_1} P_2 \xrightarrow{R_2} A_4$ and $A_2 \xrightarrow{R_1} P_2 \xrightarrow{R_2} C_2 \xrightarrow{R_3} P_3 \xrightarrow{R_4} A_4$ in an academic information network. We define different types of paths between nodes as meta paths. Formally, a meta path can be denoted as a series of different types of nodes connected by the different types of edges: $O_1 \xrightarrow{R_1} O_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} O_{l+1}$, where O_c is the node in type c and R_c is the relations between two nodes in type c. Each path has unique semantic information: the path $A \xrightarrow{R_1} P \xrightarrow{R_2} A$ means two authors are coauthors. The path $A \xrightarrow{R_1} P \xrightarrow{R_2} C \xrightarrow{R_3} P \xrightarrow{R_4} A$ means two authors published papers on the same conference.

Different from previous network embedding work which only considered the static characteristics of heterogeneous network. The goal of our work is to learn the embedding $r_v \in R^k$ of nodes in the dynamic heterogeneous information network $G = \{G^1, G^2, \dots, G^T\}$ based on meta paths. The main challenge of our work is how to preserve the historical

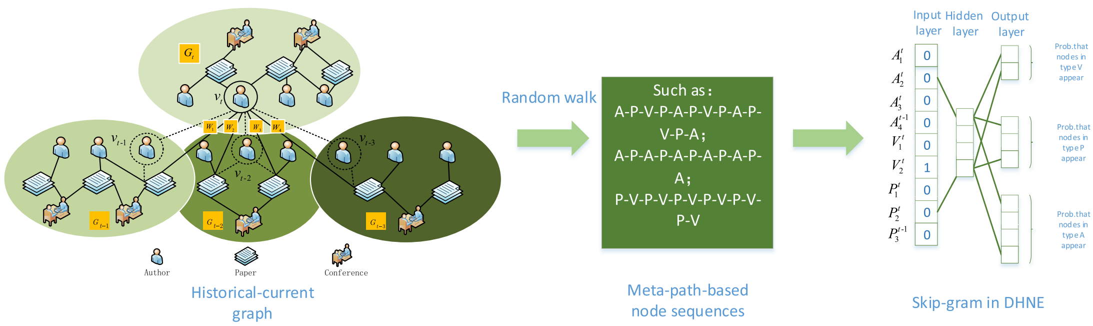


FIGURE 1. **The framework of DHNE. The framework of DHNE can be divided into three parts: (a) constructing the historical-current graph to preserve the historical information as well as current information. (b) performing the random walks under the guidance of meta path to get meta-path-based node sequences which contain both the structural and semantic information. (c) inputing the node sequences into the dynamic heterogeneous skip-gram model to learn the node embeddings.**

information as well as the current information to learn reasonable representations of nodes from the perspective of temporal and spatial condition.

IV. THE PROPOSED FRAMEWORK: DHNE

Since the heterogeneous network evolves over time, the embeddings of nodes are not only affected by the network structure at the current time, but also the historical information at the previous time steps. Hence, we propose DHNE to perform random walks on the constructed historical-current graphs in order to capture both the historical information and current information in the dynamic heterogeneous network. The framework of the proposed model DHNE is shown as Figure 1. In our work: first, we combine the historical information with the current information to construct integrated graphs named historical-current graphs. And then, we perform random walks under the guidance of meta path to get node sequences which contain semantic information. Finally, we proposed dynamic heterogeneous skip-gram model to learn the embeddings in an efficient way.

A. CONSTRUCTING THE HISTORICAL-CURRENT GRAPHS

Inspired by Word2vec model in the NLP (Natural Language Processing) and DeepWalk in homogeneous network representation learning, we regard neighborhood of nodes in the network as context of words in the text. Different from the previous work that only take current neighborhood into account, we take all of the neighborhood in the time step into account to learn the node embedding in the dynamic condition. To get all the neighbors in the time step, we construct a historical-current graph which consists of all the nodes at current time and their neighbors at the previous time steps in the time window. Based on the prior knowledge, we make the following assumptions:

- Networks always evolve smoothly over time. Therefore, when constructing a historical-current graph, nodes and its past neighbors at different times should be kept in the adjacent position in order to preserve the temporal smoothness of the network;
- Intuitively, the closer the historical neighbor node is to the current time, the greater the influence of the historical node on the current node. In order to preserve this characteristic, we construct nodes and their past neighbors in this way: the closer the historical moment is to the current time, the greater the weight of the edge between nodes and their past neighbors. The weight can be expressed as:

$$W(N_{t_j}|N_{T_i}) = \frac{exp[-(T_i - t_j)]}{\sum_{T_i-\tau<t_m<T_i} exp[-(T_i - t_m)]} \tag{1}$$

  where $W(N_{t_j}|N_{T_i})$ denotes the edge weight of node N_{t_j} and N_{T_i}, T_i denotes the current time, t_j denotes a certain historical time in the time step τ.
- The change of a node in a dynamic network is often affected by the change of its neighbors. Therefore, the influence of the historical network on the current node is mainly reflected in the local structure of the network containing the node. To capture the local information in history, we combine nth-order past neighbors of current nodes to construct the historical-current graphs. And the value of n is determined by the length of meta path we selected in the random walk process.

Based on the above assumptions, in the time step τ, we link node v_t and its past neighbors through weighted edges which can be calculated according to Equation 1. For the node at time t, a historical-current graph as shown in Figure 2 can be constructed. In order to preserve semantic information contained in the historical-current graphs, we only keep the current node v_t without retaining the historical self-nodes $v_{t-1}, v_{t-2}, \ldots, v_{t-\tau}$. For the sake of understanding, we use dashed lines to represent the historical self-nodes and the associated edges which do not actually exist in the historical-current graph.

B. RANDOM WALKS ON THE HISTORICAL-CURRENT GRAPHS

The approaches DeepWalk and node2vec have proved that random walks on the network can preserve the structural

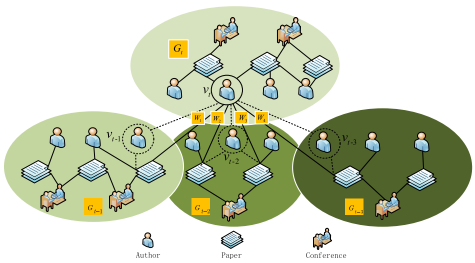
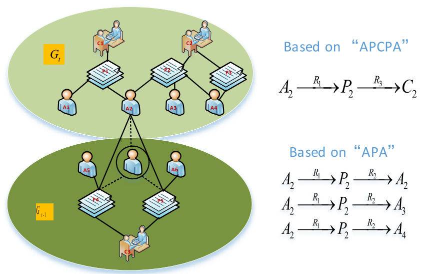


FIGURE 2. An example of the historical-current graph. The node v_t represents node in the current graph G^t and $v_{t-1}, v_{t-2}, v_{t-3}$ represent the historical self-nodes of v_t at the previous time steps. In order to preserve semantic information contained in the meta paths, we construct weighted edges directly between current node and its historical neighbors in the graph G^{t-1}, G^{t-2}, G^{t-3}.



(a) An example of historical-current graph (b) Meta-path-based random walks

FIGURE 3. An example of random walker under the guidance of different meta paths.

characteristics of network well. Therefore, in our work, we perform random walks on the historical-current graphs under the guidance of meta path to capture the structural and semantic information of dynamic heterogeneous network. If two nodes are close to each other or share many common neighbors, the probability of their co-occurrence in the random walks will be greater.

Historical-current graphs contain different types of nodes and weighted edges which lead to traditional random walk generally no longer applicable. In our work, we perform biased random walks on the historical-current graphs under the guidance of meta paths. Given a meta path M: $O_1 \xrightarrow{R_1} O_2 \xrightarrow{R_2} \ldots O_c \xrightarrow{R_c} O_{c+1} \ldots \xrightarrow{R_l} O_{l+1}$, the transition probability at step i in the random walks is defined as follows:

$$P(v_{i+1}|v_i^c, M) = \begin{cases} \dfrac{W(v_{i+1}|v_i^c)}{\sum\limits_{v_j \in H_{c+1}(v_i^c)} W(v_j|v_i^c)}, & (v_{i+1},\ v_i^c) \in E,\ f(v_{i+1}) = c+1 \\ 0, & (v_{i+1},\ v_i^c) \notin E \text{ or } f(v_{i+1}) \neq c+1 \end{cases} \tag{2}$$

where $v_i^c \in O_c$, v_{i+1} denotes the node at next step, and $H_{c+1}(v_i^c)$ denotes the neighborhood of node v_i^c in type $c+1$ which may contains neighbor nodes in the current time and historical time. We can see from the Equation 2 that only the node is a neighbor node that satisfies the type guided by meta path can become the next node that the random walk passes through. For example, in Figure 3, based on the meta path M_1: $A \xrightarrow{R_1} P \xrightarrow{R_2} C \xrightarrow{R_3} P \xrightarrow{R_4} A$, the next step of random walker on node P_2 transitioned from node A_2 must be C_2; based on the meta path M_2: $A \xrightarrow{R_1} P \xrightarrow{R_2} A$, the next step of random walker on node P_2 transitioned from node A_2 would be A_2, A_3, and A_4. Through the meta-path-based random walk, we can get node sequences containing structural correlation, semantic information and historical information on the constructed graph.

C. DYNAMIC HETEROGENEOUS SKIP-GRAM

The Skip-gram model is a language model used to maximize the co-occurrence probability between words that appear within a window in order to predict the "context" of the given word. In our work, we extend the skip-gram model to dynamic heterogeneous network representation learning. The objective of the dynamic heterogeneous Skip-gram model is to maximize the probability of observing the neighbors for a node on the historical-current graph. Given a historical-current graph which consists of nodes in $|A|$ types, the objective function can be defined as follows:

$$\max_{\theta} \sum_{v \in V} \sum_{c \in A} \log P(H_c(v)|v; \theta) \tag{3}$$

where $H_c(v)$ denotes the neighborhood of node v in type c on the historical-current graph, θ denotes the parameter set of the model. The $P(H_c(v)|v; \theta)$ can be denoted as follows:

$$P(H_c(v)|v; \theta) = \prod_{m_c \in H_c(v)} p(m_c|v; \theta) \tag{4}$$

where $p(m_c|v; \theta)$ denotes the probability of node m_c existing as the neighbor of node v. The probability is commonly defined as a softmax function. In order to normalize the specified node type and their past neighbors, we define the softmax function as follows:

$$p(m_c|v; \theta) = \frac{exp(X_{m_c} \cdot X_v)}{\sum_{u_c \in V_c} exp(X_{u_c} \cdot X_v) + \sum_{n_c \in h_c} exp(X_{n_c} \cdot X_v)} \tag{5}$$

where V_c is the set of current neighbors in type c of node v, h_c is the set of historical neighbors in type c of node v. X_{m_c}, X_v, X_{u_c} and X_{n_c} are the embeddings of nodes m_c, v, u_c and n_c, respectively. By adjusting the softmax function in the output layer of skip-gram model to adopt to the characteristics of dynamic heterogeneous networks, we can specify one set of distributions for each type of neighbors both in the history and current.

In order to alleviate the computations of our algorithm, we adopt the negative sampling method [24] to optimize our method. The probability that a node is selected as a negative sample is related to the frequency at which it appears in the node sequences. According to the work [24] of the predecessors, we select negative samples according to the degree distribution of nodes. The sampling probability can be denoted as follows:

$$p(v_i) = \frac{f(v_i)^{3/4}}{\sum_{j=1}^{K} f(v_j)^{3/4}} \tag{6}$$

Algorithm 1 DHNE

Input: The dynamic heterogeneous information network $G = \{G^1, G^2, ..., G^T\}$, time step τ, walks per node w, walk length l, embedding dimension d, negative sample number N, and neighborhood size k.
Output: The latent node embeddings $X \in R^{V \times d}$.

Initialize X
for $t = \tau - 1$ to T **do**
  $G^* =$ CreateHistorical-currentGraph(g_t,τ,$node_i$)
  for $i = 1$ to ω **do**
    for $node^*$ in G^* **do**
      Walks = BiasedRandomwalk(G^*,$node^*$,l)
      X = DynHeterogeneousSkipgram(X,k,$walks$,N)
    end for
  end for
end for

Algorithm 2 CreateHistorical-currentGraph(g_t,τ,$node_i$)

Input: graphs $g_t = \{G^t, G^t - 1, ..., G^{t-\tau+1}\}$ within the time step, time step τ, meta path type.
Output: The Historical-current graph G^*

for $node_i$ in G^t **do**
  for $j = 1$ to τ **do**
    Create weighted edges between $node_i$ and its past neighbors according to Eq.1.
    Append past subgraph of $node_i$ to Historical Graph
    Merge Historical Graph and G^t to update Historical-current Graph
  end for
end for

Therefore, the final objective function of our algorithm can be denoted as:

$$O(X) = \log \sigma(X_{m_c} \cdot X_v) + \sum_{i=1}^{N} E_{u_c^i \sim p_c(u_c), n_c^i \sim p_c(n_c)} \times [-\log \sigma(X_{u_c^i} \cdot X_v + X_{n_c^i} \cdot X_v)] \quad (7)$$

where $\sigma = 1/(1 + exp(-x))$ is the sigmoid function, N is the number of sampled negative nodes.

Commonly, we adopt Stochastic Gradient Descent method [25] to optimize the objective function in Equation 7. Algorithm 1 shows the core of our method.

V. EXPERIMENTS

In this section, we validate the effectiveness of our model on two real-world datasets. First, we introduce the datasets and baseline methods we used in our experiments in details, and then we conduct downstream tasks: node classification, visualization and trajectory analysis. Finally, we analyze the parameter sensitivity of time step. The implementation of our model DHNE and datasets are publicly available.[1]

[1] https://github.com/Yvonneupup/DHNE.git

Algorithm 3 DynHeterogeneousSkipgram(X,k,N,d)

for $v_i \in walks$ **do**
  $c = f(v_i)$
  for $v_c \in walks[i-k, i+k]$ **do**
    $X = X - \eta \frac{\partial O(X)}{\partial X}$ (According to Eq. 7)
  end for
end for

TABLE 1. Labels of DBLP dataset.

Label	Conferences
AI & ML	IJCAI, AAAI, ICML, UAI, AISTATS
AL & TH	FOCS, STOC, SODA, COLT
CV	CVPR, ICCV, ECCV, BMVC
DB	EDBT, ICDE, PODS, SIGMOD, VLDB
DM	KDD, SDM, ICDM, PKDD
IR	SIGIR, ECIR

TABLE 2. Labels of AMiner dataset.

Label	Journals/conferences
Data Mining	PKDD, SDM, KDD, WSDM
Theory	STOC, FOCS, SODA
Database	ICDE, VLDB, SIGMOD
Visualization	CVPR, ICCV, VAST, IEEE Visualization, Information Visualization
Medical Informatics	JAMIA, Artificial Intelligence in Medicine, Journal of Biomedical Informatics, IEEE Trans.Med.Imaging

A. DATASETS

1) DBLP

The DBLP dataset [26] is an author-centered academic information integration dataset composed of computer science publications, which plays an important role in the research of heterogeneous information networks. In this paper, we use a subset of the DBLP dataset which contains bibliographic information in six research areas: Artificial Intelligence & Machine Learning, Algorithm & Theory, Database, Data Mining, Computer Vision, Information Retrieval. The annual data is stored in the form of a snapshot which contains three types of nodes: papers, authors, conferences. We construct these annual snapshots from year 2000 to 2018. There are 67580 papers, 64978 authors, and 24 conferences. Table 1 shows the name of the conference and its domain.

2) AMiner

AMiner [27] is an academic search engine that helps us mine deep knowledge from academic networks. We collected a subset of AMiner dataset which contains papers published from year 1990 to 2005 in five research areas: Data mining, Theory, Database, Visualization, and Medical Informatics, Table 2 shows the detailed information. In the experimental dataset, there are 18464 papers, 23418 authors, and 19 conferences. And we use it to construct a dynamic heterogeneous

TABLE 3. Multi-class author node classification results in DBLP dataset.

Metric	Method	10%	20%	30%	40%	50%	60%	70%	80%
Micro-F1	DeepWalk	0.68121	0.71224	0.72571	0.73312	0.73812	0.73981	0.74023	0.74524
	HTNE	0.77213	0.83638	0.85747	0.87480	0.87136	0.87342	0.89354	0.89923
	Metapath2vec	0.89231	0.91383	0.94267	0.94401	0.94568	0.94562	0.95001	0.95526
	Metapath2vec++	0.91611	0.93925	0.94352	0.95336	0.95580	0.95510	0.95567	0.95568
	DHNE	**0.92809**	**0.94068**	**0.94561**	**0.95891**	**0.96523**	**0.96380**	**0.96667**	**0.96558**
Macro-F1	DeepWalk	0.67234	0.68687	0.71367	0.72546	0.72671	0.72782	0.73005	0.73423
	HTNE	0.75564	0.82894	0.84106	0.85291	0.86598	0.86221	0.87563	0.88026
	Metapath2vec	0.88564	0.90563	0.92601	0.92864	0.92897	0.93684	0.94561	0.94267
	Metapath2vec++	0.90522	0.90897	0.92567	0.92893	0.93005	0.93970	0.94126	0.94805
	DHNE	**0.91904**	**0.92857**	**0.93656**	**0.94561**	**0.95606**	**0.95672**	**0.95827**	**0.96181**

TABLE 4. Multi-class paper node classification results in AMiner dataset.

Metric	Method	10%	20%	30%	40%	50%	60%	70%	80%
Micro-F1	DeepWalk	0.71236	0.71687	0.72324	0.73869	0.74021	0.74506	0.75002	0.75896
	HTNE	0.80438	0.85504	0.87403	0.88278	0.88928	0.89363	0.89884	0.90601
	Metapath2vec	0.90012	0.91389	0.92598	0.93467	0.93897	0.94236	0.94562	0.95021
	Metapath2vec++	0.92237	0.94606	0.94997	0.95101	0.95372	0.95354	0.95414	0.95534
	DHNE	**0.94647**	**0.94900**	**0.95022**	**0.96169**	**0.96246**	**0.97057**	**0.97303**	**0.97297**
Macro-F1	DeepWalk	0.70121	0.72223	0.72545	0.73232	0.73671	0.74020	0.74521	0.75124
	HTNE	0.79213	0.84639	0.86783	0.87695	0.88286	0.88793	0.89289	0.90064
	Metapath2vec	0.90231	0.92384	0.93267	0.93898	0.94568	0.94580	0.94231	0.94560
	Metapath2vec++	0.92611	0.92926	0.94252	0.94336	0.94580	0.95010	0.95067	0.95038
	DHNE	**0.93809**	**0.94064**	**0.95141**	**0.95283**	**0.95807**	**0.96177**	**0.96235**	**0.96101**

information network which contains three types of nodes: papers, authors, conferences in 16 snapshots.

B. BASELINE METHODS

We compare our approach DHNE against four state-of-the-art methods as follows. (Since this is the first work to learn node in the dynamic heterogeneous network, we compare our method with static heterogeneous network embedding methods, homogeneous network embedding method, and dynamic homogeneous network embedding method.)

- DeepWalk [6]: DeepWalk is a homogeneous network representation learning method which learns node embeddings in static graphs. In our experiment, we run DeepWalk on every snapshot and use the unified embeddings to do the comparative experiments.
- HTNE [28]: HTNE performs the dynamic homogeneous network representation learning based on the Hawkes process which can capture both the historical and current information from the perspective of temporal sequences.
- Metapath2vec [4]: Metapath2vec conducts random walks in heterogeneous information network under the guidance of meta paths, and then adopts the Skip-gram model to learn the node representations.
- Metapath2vec++ [4]: Metapath2vec++ improves the normalization of the softmax function in metapath2vec to separates different types of nodes in the output layer.

We set the default parameters of our method DHNE as follows:

(1) The vector dimension: 128;
(2) The negative sample number: 5;
(3) The gradient drop learning rate: 0.01;
(4) The random walk length: 100;
(5) The number of walks per node: 50;
(6) the neighborhood size: 7;
(7) time step: 4.

All the baseline methods keep the same parameter to DHNE as much as possible. The parameters in the baseline algorithm that do not belong to the above parameters are set following the suggestion in their original papers.

C. DOWNSTREAM TASKS

In this section, we carry out network application tasks such as node classification and visualization to verify the feasibility and effectiveness of our method DHNE.

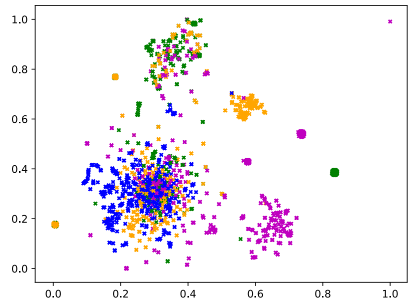
1) CLASSIFICATION

We conduct node classification task on DBLP and AMiner datasets, and we get labels of nodes used in the classification experiment in this way: the labels of conference are shown in Table 1 and Table 2, the labels of papers are determined by the conference to which the paper belongs, and the labels of authors are determined by the papers published by the authors. So, for DBLP dataset, we can divide each type of nodes into six categories. For AMiner dataset, nodes can be divided into five categories. In our experiment, the embeddings learned from different methods were classified by a linear SVM classifier. We repeat classification experiment ten times and take the average of Micro-F1 and Macro-F1 scores as the final classification results.

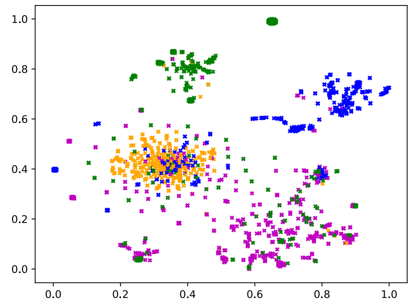
We set the training set size varying from 10% to 80%. Table 3 demonstrates the the average classification results of authors from 2000 to 2010 on DBLP dataset. Table 4 demonstrates the classification results of papers from 1990 to 2000 on AMiner dataset. We can see that, compared with

TABLE 5. Trajectory classification of authors on DBLP and AMiner dataset.

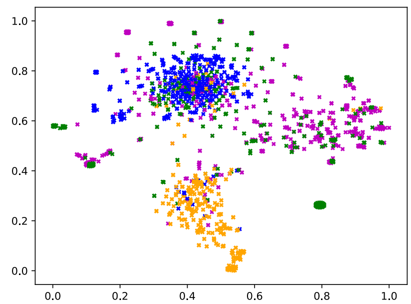
Dataset	Method	Accuracy	Micro-F1	Macro-F1
DBLP	DeepWalk	0.7132	0.7067	0.7034
	HTNE	0.8048	0.7950	0.7913
	Metapath2vec	0.8206	0.8159	0.8108
	Metapath2vec++	0.8258	0.8176	0.8127
	DHNE	**0.8521**	**0.8461**	**0.8435**
AMiner	DeepWalk	0.7324	0.7283	0.7261
	HTNE	0.8192	0.8104	0.8049
	Metapath2vec	0.8356	0.8297	0.8261
	Metapath2vec++	0.8367	0.8268	0.8256
	DHNE	**0.8763**	**0.8657**	**0.8611**



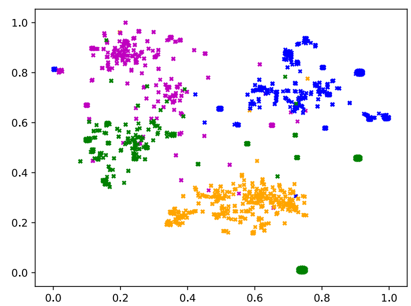
(a) Deepwalk.



(b) HTNE.



(c) Metapath2vec++.



(d) DHNE.

FIGURE 4. Visualization of authors from four research areas in Aminer dataset.

baseline algorithms, the DHNE algorithm proposed in this paper performs better than all the baseline methods on metrics Macro-F1 and Micro-F1. On the DBLP dataset, when given 80% of nodes as the training set, the DHNE algorithm achieves 1.70% ∼ 22.82% improvements on Macro-F1 score. When the training set takes up 70%, the DHNE algorithm gets the highest Micro-F1 score value, which achieve 1.11% ∼ 22.64% improvements comparing with the baseline methods. On the AMiner dataset, when the training set takes up 70%, the DHNE achieves 1.17% ∼ 21.69% improvements on Macro-F1 score and 1.89% ∼ 22.30% improvements on Micro-F1 score comparing with the baseline algorithms.

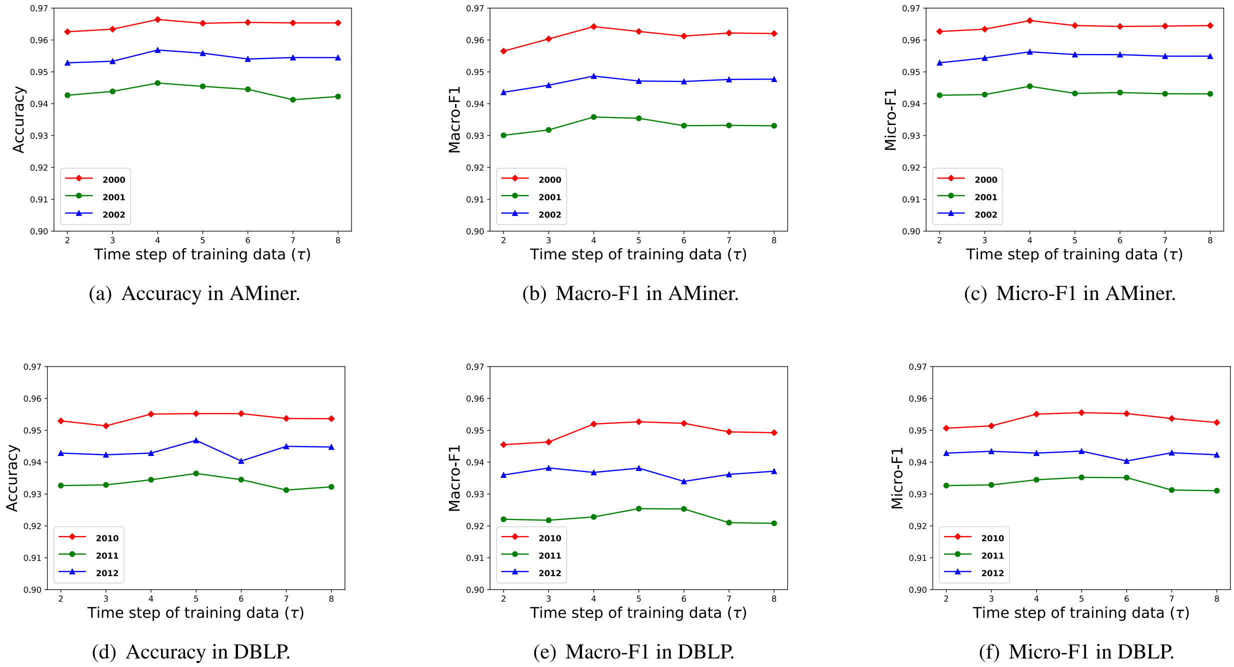
From the experimental results, we can see, by constructing the historical-current network graphs, we can fuse the historical information into the representations of nodes which can help us to improve the classification accuracy. Therefore, compared with baseline methods which only take the current information into account, our method performs better.

2) ANALYZE THE TRAJECTORY OF AUTHORS

Heterogeneous network representation learning methods on static graphs tend to focus on mining information in static condition. When analyzing the dynamic characteristics of networks, the performance of these algorithms will be greatly reduced. Our method learns the node embeddings in dynamic condition which can help us analyze dynamic characteristics of nodes such as node trajectory.

In the academic information networks, we can know whether the author is "Specific Researcher" or "Interdisciplinary Researcher" through the trajectory of the author over a period of time. For authors in the temporal academic networks, authors may research in different fields and we can call them "Interdisciplinary Researcher", while authors research in a certain field can be called "Specific Researcher". From the labels of authors in different years, we can get the category to which the author's trajectory belongs. For a period of time, if the author's label remains the same, then the author's trajectory belongs to "Specific Researcher". While if the author's label changes, the author's trajectory belongs to "Interdisciplinary Researcher". We can get author embeddings in every year, thus we can perform the trajectory classification for authors in the time step in order to mine the dynamic information about the network. Table 5 demonstrates the trajectory classification results on DBLP dataset from year 2010 to 2018 and on AMiner dataset from year 1995 to 2005.

We can see, the proposed method DHNE have got better performance than baseline methods. On DBLP dataset, DHNE achieves 2.63% ∼ 13.99% improvements in accuracy, 2.85% ∼ 13.94% improvements in Micro-F1 score, 3.08% ∼ 14.01% improvements in Macro-F1 score.



(a) Accuracy in AMiner. (b) Macro-F1 in AMiner. (c) Micro-F1 in AMiner.

(d) Accuracy in DBLP. (e) Macro-F1 in DBLP. (f) Micro-F1 in DBLP.

FIGURE 5. **Impacts of time step on DBLP and AMiner dataset.**

On AMiner dataset, DHNE achieves 3.96% $\sim$ 14.39% improvements in accuracy, 3.89% $\sim$ 13.74% improvements in Micro-F1 score, 3.55% $\sim$ 13.50% improvements in Macro-F1 score. Therefore, on both of the dataset DBLP and AMiner, DHNE achieves great improvements in node trajectory classification which indicates that the historical information combined in the current network improves the quality of node embeddings for analyzing the dynamic characteristics of nodes.

3) VISUALIZATION

Visualization is an effective and intuitive downstream task to evaluate the quality of node embeddings learned from different approaches. We leverage the t-SNE algorithm to visualize the representation vectors of 2663 authors from four fields (Data Mining, Theory, Database, Visualization) in 2005 on the AMiner dataset into the 2-dimensional space. We use different colored dots to represent authors in different research areas. Specifically, orange dots represent authors in "Data Mining", green dots represent authors in "Theory", purple dots represent authors in "Database", blue dots represent authors in "Visualization". Figure 4 shows the visualization results of node embeddings obtained by different algorithms.

As can be seen from the Figure 4, the DeepWalk algorithm can not map the authors from four fields to independent communities, they are totally confused; The HTNE algorithm can map authors in "Data Mining" domain into an independent community, but it maps authors in the "Database" and "Theory" domain to relatively scattered locations, failing to preserve the properties of nodes in this two research areas; The metapath2vec++ algorithm can map authors in "Visualization" and "Data Mining" domains to relatively independent communities, but it fails to separate authors in "Theory" and "Database" domains completely; Compared with the baseline algorithms, the proposed DHNE algorithm can map authors into different communities and there are clear margins among different areas. The visualization results indicate that the historical information combined with current information in our method can help us do community detection [29]. This is because the formation of a community is often related to historical information, which can assist us in discovering communities. The embeddings generated by our method DHNE integrate historical information and current information, which can preserve the community information better. Therefore, the embeddings learned by our method DHNE perform better than other baseline methods in visualization.

D. PARAMETER SENSITIVITY

In this section, we analyze the parameter time step T, which determines the historical information contained in the constructed historical-current graph. The larger the parameter T, the more historical information the historical-current graph contains. We analyze the classification results of authors from 2000 to 2002 in the AMiner dataset and authors from 2010 to 2012 in the DBLP dataset based on DHNE with T varying from 2 to 8 to validate the parameter sensitivity of our method.

As the Figure 5 shown, on the AMiner dataset, when the time step is set to 4, we can obtain the maximum value of Accuracy, Macro-F1 and Micro-F1 score. When $1 < T < 4$, the larger the parameter T, the higher the scores. And when $T > 4$, the Accuracy, Macro-F1 and Micro-F1 scores are relatively stable when the parameter T increases. We can analyze the experimental results theoretically: when the time step is short, the constructed historical-current graph cannot adequately capture the historical information affecting the node representations, which results in low classification accuracy; when the time step is too large, it will only lead to an increase in computation rather than improving the performance of the node representations in classification accuracy. Because the farther the historical information is from the current time, the less influence it will have on the current node representations. Therefore, the most appropriate time step should be a moderately sized value, rather than simply choosing a large time step T to integrate more historical information.

Besides, we can see from the Figure 5 that on different datasets, time step have different influence on classification results. For DBLP dataset, when the parameter T is set to 5, the Accuracy, Macro-F1 and Micro-F1 scores are largest. Intuitively, there are two reasons for this difference: first, the authors from the two datasets are selected from different years. We select authors in AMiner dataset from 2000 to 2002 while selecting authors in DBLP dataset from 2010 to 2012, the connection between authors in the early years may change faster than authors in recent years; Secondly, the authors from the two datasets are selected from different research areas. The connection between authors in the research areas included in the Aminer dataset may change faster than authors in research areas from DBLP dataset. Since we leverage the effective historical information to assist the node embedding, the faster the connection between authors changes, the smaller the time step should be selected. Therefore, the most appropriate time step T on AMiner dataset is smaller than the most appropriate time step T on DBLP dataset.

VI. CONCLUSION

In this paper, we proposed DHNE, a novel dynamic heterogeneous network representation learning method to mine the rich information in history as well as information at current time. By constructing the historical-current graphs in the time step, we can combine the historical and current information in the original network. Through random walks on the constructed historical-current graphs under the guidance of meta path, we can capture the semantic information in the network. Since we integrated multiple types of information into network representation learning, the embeddings learned by our model performed well in the downstream tasks such as node classification and visualization.

At present, the research on network dynamic characteristics is still in its infancy, and there are very few researches on dynamic heterogeneous information networks. The integration of dynamic information makes the network more complex, and dynamic networks often have a very large scale. Thus, there are some challenges in future work: how to process large-scale networks efficiently, how to integrate the attributes of nodes into network representation learning, and how to process streaming data in real time are the next research focus.

REFERENCES

[1] C. Luo, R. Guan, Z. Wang, and C. Lin, "HetPathMine: A novel transductive classification algorithm on heterogeneous information networks," in *Proc. Eur. Conf. Inf. Retr.*, Cologne, Germany, 2014, pp. 210–221.

[2] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: Meta path-based top-K similarity search in heterogeneous information networks," *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[3] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han, "Co-author relationship prediction in heterogeneous bibliographic networks," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining*, Kaohsiung, Taiwan, Jul. 2011, pp. 121–128.

[4] 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*, Halifax, NS, Canada, 2017, pp. 135–144.

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

[6] 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*, New York, NY, USA, 2014, pp. 701–710.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efcient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Represent.*, Scottsdale, AZ, USA, 2013, pp. 1–12.

[8] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery data Mining*, New York, NY, USA, 2016, pp. 855–864.

[9] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2016, pp. 1225–1234.

[10] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, and M. Guo, "GraphGAN: Graph representation learning with generative adversarial nets," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, 2018, pp. 2508–2515.

[11] Q. Dai, Q. Li, J. Tang, and D. Wang, "Adversarial network embedding," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, 2018, pp. 2167–2174.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 2672–2680.

[13] J. Li, H. Dani, X. Hu, J. Tang, Y. Chang, and H. Liu, "Attributed network embedding for learning in a dynamic environment," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Singapore, 2017, pp. 387–396.

[14] D. Zhu, C. Peng, Z. Zhang, P. Jian, and W. Zhu, "High-order proximity preserved embedding for dynamic networks," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 11, pp. 2134–2144, Nov. 2018.

[15] L. Du, Y. Wang, G. Song, Z. Lu, and J. Wang, "Dynamic network embedding: An extended approach for skip-gram based network embedding," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, 2018, pp. 2086–2092.

[16] G. H. Nguyen, J. B. Lee, R. A. Rossi, N. K. Ahmed, and S. Kim, "Continuous-time dynamic network embeddings," in *Proc. Companion Web Conf. Int. World Wide Web Conf.*, Lyon, France, 2018, pp. 969–976.

[17] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 833–852, May 2018.

[18] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl.-Based Syst.*, vol. 151, pp. 78–94, Jul. 2018.

[19] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Eng. Bulletin.*, vol. 40, no. 3, pp. 52–74, 2017.

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

[21] S. Y. Chang, W. Han, J. Tang, G. J. Qi, C. C. Aggarwal, and S. T Huang, "Heterogeneous network embedding via deep architectures," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Sydney, NSW, Australia, 2015, pp. 119–128.

[22] C. Shi, C. Zhou, X. Kong, P. Yu, G. Liu, and B. Wang, "HeteRecom: A semantic-based recommendation system in heterogeneous networks," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Beijing, China, 2012, pp. 1552–1555.

[23] T. Chen and Y. Sun, "Task-guided and path-augmented heterogeneous network embedding for author identification," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Cambridge, U.K., 2017, pp. 295–304.

[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, CA, USA, 2013, pp. 3111–3119.

[25] L. Bottou, "Stochastic gradient learning in neural networks," *Proc. Neuro-Nımes*, vol. 91, no. 8, p. 12, 1991.

[26] C. Moreira, P. Calado, and B. Martins, "Learning to rank academic experts in the DBLP dataset," *Expert Syst.*, vol. 32, no. 4, pp. 477–493, 2015.

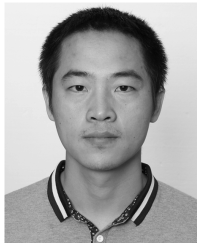
[27] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Las Vegas, NV, USA, 2008, pp. 990–998.

[28] Y. Zuo, G. N. Liu, H. Lin, J. Guo, X. Hu, and J. Wu, "Embedding temporal network via neighborhood formation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., 2018, pp. 2857–2866.

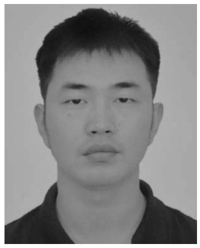
[29] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010.

YING YIN was born in Mianzhu, Sichuan, China, in 1994. She received the bachelor's degree from Xi'an Jiaotong University, in 2017. She is currently pursuing the master's degree with Information Engineering University. Her current research interests include data mining and network representation learning. Her awards and honors include the National Encouragement Scholarship and the First Prize in the Mathematical Modeling Contest.

LI-XIN JI was born in Huai'an, Jiangsu, China, in 1969. He is currently the Chief Engineer of the China National Digital Switching System Engineering Technological Center. His current research interests include data mining and telecommunication network information security.



JIAN-PENG ZHANG was born in Langfang, Hebei, China, in 1988. He received the master's degree from Information Engineering University, in 2014, and the Ph.D. degree from the Eindhoven University of Technology, in 2018. He is currently an Assistant Researcher with Information Engineering University. His current research interests include data mining, big data analytics, and social network analysis.



YU-LONG PEI was born in Langfang, Hebei, China, in 1991. He is currently pursuing the Ph.D. degree with the Department of Mathematics and Computer Science, Eindhoven University of Technology. His current research interests include social network analysis, text mining, and machine learning.