Task-Guided Context-Path Embedding in Temporal Heterogeneous Networks

QIAN HU\textsuperscript{1,2}, FAN LIN\textsuperscript{1}, BEIZHAN WANG\textsuperscript{1}, AND CHUNYAN LI\textsuperscript{3}

\textsuperscript{1}School of Informatics, Xiamen University, Xiamen 361005, China
\textsuperscript{2}School of Media and Communications, Guizhou Normal University, Guiyang 550000, China
\textsuperscript{3}Graduate School, Yunnan Minzu University, Kunming 650504, China

Corresponding author: Fan Lin (iamafan@xmu.edu.cn)

ABSTRACT Network embedding maps the nodes of a network to a continuous vector space, which can then be used as the input to downstream tasks, such as node classification, node clustering, link prediction, and similarity search. To learn network embedding more effectively, many technologies adopt the approach of random walk to obtain the network structure. As the meta-path of heterogeneous networks emerges, network embedding will be equipped with more semantic interpretation. Consequently, various random walks, based on meta-path strategies, have been proposed for network embedding. However, the combination of semantic and structure in a heterogeneous network cannot achieve ideal results. To overcome this challenge, we start from a task-guided issue by combining the timestamps information in the heterogeneous network, and then employing the method of temporal segmentation to decompose the network into a continuous temporal sequence. Finally, the set of context-paths between nodes is calculated in a continuous vector by the depth-first meta-path search algorithm. More precisely, we propose a Temporal Sliding Density Walk (TSDW) algorithm by combining network semantics and structure effectively. Empirical results for network data show that TSDW could significantly outperform the state-of-the-art representation learning models, including DeepWalk, LINE, Node2vec, PTE, Meapath2vec, HIN2vec, HTNE, and CTDNE by 3.02\% to 44.9\% of Macro-F1, 0.9\% to 18.92\% of Micro-F1 in multi-class node classification and 21\% to 47\% of NMI in node clustering.

INDEX TERMS Context-path, network embedding, temporal heterogeneous network.

I. INTRODUCTION

An increasing number of networks, such as literature and film-rating networks, are being represented as heterogeneous due to a deeper understanding regarding real-world networks among researchers. Although this kind of network representations make their structures more complex, they carry abundant semantic information. Furthermore, apparent timestamps information of these real-world networks (such as publication year of paper nodes and timestamp information of film rating) is universal. Reasonable use of such timestamps information to network embedding has drawn the attention of researchers around the world globally.

The process of embedding a heterogeneous network based on a time series is as follows. Firstly a dynamically developed network structure is split into several consecutive snapshots. And then relationships among them are found to simplify the further network embedding tasks (known as the graph embedding or network representation learning) \cite{1}. Since a time series is continuous, the segmentation process of a time series will inevitably lead to the discretization of temporal information in networks, it is undeniable that a snapshot is more power interpretable in certain time windows \cite{2}. In reality, a network is formed by continuously adding nodes and edges, which can be regarded as a dynamic process driven by the interacting events between the nodes and their neighbors. Therefore, the neighborhoods of different nodes are not formed at the same time, and the observed snapshot structure of the network is the product of neighborhoods accumulation in a certain time period. Furthermore, a reasonable temporal segmentation is more conducive for restoring semantic information in a heterogeneous network. And to some extent, it is possible to eliminate redundant noise interference. Hence, it is necessary to adopt an effective means of decomposing a network through temporal sequence segmentation and temporal sequence sliding.
The most essential goal in embedding a traditional network is to map the nodes (or links) of a heterogeneous network to a continuous low-dimensional vector space, usually the random walk method is used [3]. This approach collects neighboring nodes randomly (or restrictedly) and establishes a long enough walk path (node sequence) to obtain as much topology of a network as possible. Furthermore, to represent rich semantic interaction related information in heterogeneous networks, meta-path [4] is also often used as a constraint to the random walk [5], [7], [8]. The significance of the meta-path is reflected on the semantic relationship between different meta-paths of the same pair of nodes, and often there is also a manifestation of semantic relationship between the nodes. This approach is widely used in classification [6], clustering [9], and link prediction [10]. Many works use expert-directed meta-paths as the basis for network embedding [5], [7], [8], [11], [21]. In these works, the meta-paths are usually made shorter to mine the obvious and strong semantic relationship in the network [21]. In fact, due to the sparsity of the real-world networks, similar or related nodes may not share direct neighbors (even do not share second-order or higher-order neighbors). In communication science, longer meta-paths may indicate similarity and correlation of nodes spreading to further neighbors [12]. Effective use of this communication capability can solve the sparsity of networks and greatly improve the network coverage rate. For capturing the semantics behind the pairwise relationship, we propose to encode the paths explicitly between a pair of nodes, called context paths.

Furthermore, this study attempts to solve some task-guided problems, such as ‘how to embed the output of scholars in a given domain?’. Accordingly, we propose a heterogeneous network embedding strategy using Temporal Sliding Density Walk Strategy (TSDW), rather than embedding with random walks based on expert-directed meta-paths. Our strategy works in two steps: 1) efficient temporal segmentation and time window sliding, which assume that the nodes and links are collected to express the strongest semantic connection in a certain period; 2) we attempt to combine the semantics with the structure information in the snapshot by using a density walk. The contributions of this work are highlighted below:

- We proposed the temporal segmentation and time window sliding strategy on a heterogeneous network to rationally utilize the dynamic evolution process of the network structure.
- We proposed TSDW, an algorithm that can automatically extract meta-paths, and provided the task-guided context-paths embedding.
- We conducted experiments with a real-world academic dataset, and then compared the results with those of the state-of-art embedding algorithms used for classification and clustering tasks. The results show that our algorithm performs well in the specific task-guided mission.

II. RELATED WORKS

Existing network embedding methods can be divided into two main categories: homogeneous networks and heterogeneous networks. In homogeneous networks, all nodes are of the same type, such as a social network whose all nodes are users. The corresponding network embedding technology either attempts to use the adjacency matrix of the network for matrix decomposition [13] or adopts the random walk to extract the node sequence and then feeds it into a SkipGram model [14] to get the node embeddings [15], [16], [24]. The SkipGram model learns the node embeddings by maximizing the probability of co-occurrence of nodes in a particular context window. These methods are applied to practical applications due to their good scalability. On the other hand, heterogeneous networks are widely studied as different types of nodes, and links exist there, which can better express the semantic relations among nodes. Recent studies [5], [7], [8], [11], [21] show that the performance of the network embedding can be significantly improved by considering the nodes’ heterogeneity and the network links. Therefore, the subject of network embedding is more inclined to heterogeneous networks. PTE [17] assumed that the heterogeneous networks can be decomposed into several bipartite networks with only two types of nodes and then summed up the respective loss functions in the embedding and prediction tasks.

In order to gain rich semantics in the heterogeneous network, meta-paths are often used as guidelines for applying random walk in the network embeddings. For example, in Metapath2vec [7], ‘Author-Paper-Author’ and ‘Author-Paper-Venue-Paper-Author’ are selected manually as meta-path guidelines to limit the random walk. Shi et al. [21] used a set of predefined meta-paths to guide the random walk in generating node embeddings, merged with the item recommendation task. Chen et al. [5] proposed a path enhanced heterogeneous network embedding method intended for author identification task and performed under blind review settings. It relies on a specific task to select meta-paths and gradually combines them into the learning process. HIN2vec [8] utilized different types of relationships among nodes to guide random walks by combining meta-paths of a certain length. However, the study in [11] showed that the performance of network embedding is sensitive to the selected meta-paths.

Several other methods have been proposed for embedding networks with timestamps. Nguyen et al. [18] proposed a continuous dynamic embedding method based on random walk, whose path can move only in the forward direction. Zuo et al. [33] adopted Hawkes [34] process to maintain the network structure in embedding by capturing the changes of nodes and edges in a dynamic network. Recently these algorithms [19], [20] have been applied to obtain more meaningful embeddings from a series of discrete static snapshots. However, the research subject of these methods is a homogeneous network, which lacks semantic extraction. More surveys and applications can be found in [1], [23], [28]–[31].
III. PRELIMINARIES

In this section, we first define the temporal heterogeneous network and other terminologies related to the present study, and then show their notations and assumptions.

Definition 1 (Temporal Heterogeneous Network): A heterogeneous network is a graph $G = (\mathcal{V}, \mathcal{E})$ with node mapping as $\varphi : \mathcal{V} \rightarrow \mathcal{T}$ and edge mapping as $\psi : \mathcal{E} \rightarrow \mathcal{R}$, where $|\mathcal{T}| > 1$ is the number of types of nodes and $|\mathcal{R}| > 1$ is the number of types of edges. For a temporal heterogeneous network, consider $t_\mathcal{V}$ is the timestamp mapping of a node which maps a subset of nodes $\mathcal{V}_T \in \mathcal{V}$ and $t_\mathcal{E}$ is the timestamp mapping of an edge which maps a subset of edges $\mathcal{E}_T \in \mathcal{E}$. We define $\mathcal{V}_T$ as the set of all the nodes having timestamps and $\mathcal{E}_T$ as the set of all the edges having timestamps, where $\mathcal{V}_T$ and $\mathcal{E}_T$ will be empty sets if none of the nodes and edges of a heterogeneous network will have any timestamp.

Example 1: Fig. 1(a) shows a temporal heterogeneous network consisted of four different types of nodes (author (A), paper (P), venue (V), domain (D)), and four different types of edges (A $\leftrightarrow$ P: an author writes papers or a paper is written by an author, P $\leftrightarrow$ D: a paper belongs to a domain or a domain contains a paper, P $\leftrightarrow$ V: a paper is published in a venue or a venue publishes a paper, P $\leftrightarrow$ P: a paper cites another paper or a paper is cited by another paper). Additionally, a P-type node has a timestamp (publishing year) on it. $t_0, t_1, t_2, t_3$ represent the years of publication of the papers.

Definition 2 (Meta-Path): Considering $A_i \in \mathcal{V}$ and $R_l \in \mathcal{E}$ in a heterogeneous network, the meta-path [4] $\rho$ can be defined as a path in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{l-1}} A_l$ (abbreviated as $A_1A_2 \cdots A_l$). It describes the composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between $A_1$ and $A_l$, where $\circ$ denotes the composite relationship between the two nodes. The meta-path contains rich semantics.

Example: Fig. 1(b) shows three different paths connecting three authors in the AI domain: Domain-Paper-Author (D-P-A), Domain-Paper-Paper-Author (D-P-P-A), and Domain-Paper-Venue-Paper-Author (D-P-V-P-A). Each meta-path expresses specific semantics. For example, DPA indicates that the paper of the author belongs to the domain, DPVA indicates that the paper cited by the author’s paper belongs to the domain, DPVPA indicates that the paper from the venue in which the author’s paper was published belongs to the domain. Note that no symmetry is enforced in the definition of the meta-path.

Definition 3 (Context-Path): For given two nodes $v_i, v_j \in \mathcal{V}$, a set of context-paths from $v_i$ to $v_j$ for all paths $\omega \in \mathcal{V}_T^\mathcal{P}$ are denoted by $\mathcal{C}(v_i, v_j)$. A set of context-paths between two given nodes contains all meta-paths between them.

Example: Fig. 2 shows all the context-paths from the data-mining domain to the author ‘Alice’ as shown in Fig. 1(a), where each context-path corresponds to a semantic rich meta-path.

Note that the meta-path ‘D-P-P-A’ is not unique. Many context-paths sharing the same type of meta-paths in the real-world data set (such as ACM, 1 DBLP2).

Finally, the notations and explanations used in our study are shown in Table 1. The next section presents TSDW in detail.

IV. TEMPORAL HETEROGENEOUS NETWORK EMBEDDING WITH TSDW

In this section, we introduce our algorithm TSDW, which is a strategy for embedding a heterogeneous network using temporal segmentation and time window sliding strategies, rather than using random walks based on expert-directed meta-paths. Firstly, we present the assumptions and problem definitions, and then we introduce our technology in detail.

We assume that context-paths with closed temporal relationships have stronger semantic relationships. For example, it is more likely that two authors closely collaborate under the condition that they simultaneously publish two papers.

1http://dl.acm.org/
2https://dblp.uni-trier.de
in a relatively shorter time period. We assume that stronger semantic relationships are reflected in the structure density of the network. As an example, if the author ‘Alice’ had made a great contribution to the domain of data mining, many paper nodes will link to this author node, and her (his) papers will also cite more papers. There will also be a lot of venue nodes and domain nodes linked to these papers. We also assume that although the semantic strength of a longer meta-path is reduced [4], it reflects the structure information of the network better as the path contains more types of nodes. Based on the above assumptions, we propose the problem described below.

**Problem Description:** In a temporal heterogeneous academic network, we attempt to provide the embedding of the authors who are related to a particular domain. With the aforementioned notations and definitions, the embedding of the task-guided context paths to be addressed is formally defined as follows:

**Input:** For a specific node $v_{source}$ and a set of same type of nodes $V_{targetnodes}$, this technique extracts associated set of context-paths $C_{P_s \rightarrow t}$ from $v_{source}$ to every node in $V_{targetnodes}$.

**Output:** Present a novel context-path embedding $C_{P_s \rightarrow t}$ and evaluate it by classification and clustering tasks.

In the following subsections, we present our proposed temporal segmentation and sliding process, followed by embedding learning with a density walk strategy. Fig. 3 shows the framework of our TSDW strategy.

### A. TEMPORAL SEGMENTATION AND SLIDING

As mentioned above, the interaction of nodes in a special time window reflects the interaction of events in that period. Moreover, temporal information is ubiquitous in heterogeneous networks. Since closely interacting temporal events can better reflect accurate semantic than those with long temporal interaction, we believe that appropriate use of temporal information can greatly improve the interpretability of network embedding. For example, in an academic network, recent co-authors likely have a stronger academic relationship than those co-authors who collaborated decades ago. Hence, it is of interest to study the problem of temporal segmentation in a temporal heterogeneous network.

However, the temporal space is sequential and continuous, and hence the segmentation method will inevitably lead to information loss and fragmentation [2]. Also, for pragmatic use, the attempt to incorporate time in the embedding of a heterogeneous network should be accompanied by practical algorithms. Hence, a strategy of time window sliding is employed to sample the temporal space. Fig. 4 shows our time window sliding strategy.

The flexibility of temporal segmentation is related to the fact that a larger window on the excessively sparse network can be adopted, and by narrowing the time window in the network with rich links, the complexity of the embedding
algorithm can be reduced. The major contribution of the temporal window sliding is to reflect the structural evolution of the temporal heterogeneous network at a minimum time granularity. Note that the timestamp information may be associated with the nodes of the network (as a node attribute), or may appear on the links between nodes (as the action of event occurrence). In this study, we focus on the timestamps associated with nodes (indeed publishing time of papers).

B. EMBEDDING WITH TSDW

The essence of embedding learning represents the nodes (or links) of a network using a continuous low-dimensional vector space. The key idea of TSDW is started with the connectivity between nodes according to different task-guided requirements and represent the connectivity as a continuous vector. For every snapshot segmentation (refer to Section IV-A), the TSDW algorithm is divided into two parts: 1) depth-first meta-path search and 2) context-path density calculation.

1) DEPTH-FIRST META-PATH SEARCH

Unlike the random walk-based algorithms that consider node sequences as sentences, we attempt to perform the weighting between them. Hence, we propose the Depth-First Meta-path Search in the snapshot sub-network. The benefit of temporal segmentation is that the complexity of the DFS (Depth-First Search) is reduced greatly. To ensure reasonable computational complexity of DFS, additionally, we set a maximum path length of context-path $L_{\text{max}}$, which means no path in the set of context-paths will exceed it. Naturally, after executing the DFS, we will get a set of context-paths from the source node to the target node in the snapshot segmentation. It is worthy to note that each traverse path between two nodes is a kind of meta-path containing some rich semantics or a structure. Algorithm 1 illustrates the generation of the set of context-paths.

For each path $w$ in the set of DFS paths $W$, we pick out the sequence of types, i.e., meta-path $\rho$, and ultimately all kinds of meta-paths and DFS paths between given nodes are extracted in $C$.

2) CONTEXT-PATHS’ DENSITY CALCULATION AND EMBEDDING

As shown in Fig. 2, for a given pair of nodes $(v_{\text{data-mining}}, v_{\text{Alice}})$ on a task-guided depth-first meta-path search, there exists a set of context-paths $C^P$, i.e., a set of node sequences between domain type ‘data-mining’ and author type ‘Alice’. The set of meta-paths $P$ is extracted in an unsupervised way. We assume that stronger semantic relationships are reflected in the structure density of the network. Obviously, if ‘Alice’ is a prolific author who is interested in the data-mining domain, there would be more traverse paths in $C^P$. In short, the context paths from $v_{\text{data-mining}}$ to $v_{\text{Alice}}$ reveal the output level of the author ‘Alice’ in the ‘data-mining’ domain.

More precisely, given a pair of nodes $(v_i, v_j)$ and the set of context-paths $C^P_{i\rightarrow j}$ associated with their depth-first meta-path search, we consider the transition probability of the type between two nodes $(v_k, v_{k+1})$ of any single traverse path in $C^P_{i\rightarrow j}$, i.e., the transition probability of $A_k \xrightarrow{R_{i\rightarrow j}} A_{k+1}$ (abbreviated as $A_kA_{k+1}$). A reasonable attempt would be to measure the proportion of $A_{k+1}$ from the neighbors of the current node $v_k$. The transition probability can be expressed by (1).

$$p(A_kA_{k+1}|v_k;\rho) = \begin{cases} \frac{|N^{A_{k+1}}(k)|}{|N_k|}, & \rho(v_k, v_{k+1}) \in R \text{ and } \rho \in C^P_{i\rightarrow j} \\ 1, & v_k = v_j \end{cases}$$  (1)

where $N^{A_{k+1}}(k)$ is the set of the first-order neighbors of node $v_k$ of type $A_{k+1}$. However, in the heterogeneous network, the proportions of different types of nodes are different. For example, in an academic network, the number of types of authors must be smaller than the number of types of papers, while the magnitudes of the venue types and paper types are not of the same order. More precisely, (1) can be rewritten as in (2).

$$p(A_kA_{k+1}|v_k;\rho) = \begin{cases} 0, & G^{(\cdot)} = \emptyset \\ \frac{|N^{A_{k+1}}(k)|}{|N_k|} \times \frac{|G^{(\cdot)}(v_{k+1})|}{|G^{(\cdot)}(v)|}, & \rho(v_k, v_{k+1}) \in \mathcal{E} \text{ and } \rho \in C^P_{i\rightarrow j} \\ 1, & v_k = v_j \end{cases}$$  (2)

where $G^{(\cdot)}(v_{k+1})$ represents the nodes set of $v_{k+1}$ type in each snapshot of the temporal heterogeneous network $G$, and $G^{(\cdot)}(v)$ is the nodes set in a snapshot. The context-path density ratio between the two nodes $(v_j, v_j)$ is calculated using (2) by considering their types. Then, equation (2) can be extended as in (3).

$$p(A_{i\rightarrow j}(v_j;\rho) = \prod_{i\rightarrow j} p(A_kA_{k+1}|v_k;\rho)$$  (3)

The result of equation (3) is the density probability of each traverse path in $C^P_{i\rightarrow j}$.

As shown in Algorithm 1 that each meta-path $\rho$ in $C^P_{i\rightarrow j}$ corresponds to multiple traverse paths, the density probability between $(v_i, v_j)$ of each meta-path finally can be expressed by (4).

$$p(\rho_{i\rightarrow j}(v_i,v_j);P) = \sum_{w \in \mathcal{W}} \prod_{i\rightarrow j} p(A_kA_{k+1}|v_k;\rho) \quad \mathcal{P}, \mathcal{W} \in C^P_{i\rightarrow j}$$  (4)
where \( w \) is a traverse path belonging to the set of traverse paths \( \mathcal{W} \) whose types of node sequences correspond to meta-path \( \rho \). In summary, when a source node and a target node are given, there exist a set of multiple meta-paths \( \mathcal{P} \). In our study, we extract data of 20 years. The extracted dataset contains 4,348 authors, 98 venues, and 113,927 papers. We choose the ‘Artificial Intelligence (AI)’ domain as the source node and all the author nodes as the target nodes. That is, in each snapshot, the depth-first meta-path

\[
\begin{align*}
\rho &= \rho_1 \\
&= \{1 \rightarrow 4 \rightarrow 5 \rightarrow 9\} \\
&= \{1 \rightarrow 6 \rightarrow 5 \rightarrow 9\}
\end{align*}
\]

are the same length. Fig. 5 demonstrates the calculation process in a simple heterogeneous network.

Note that the arrows of the links do not indicate a directed network, but merely indicate the traverse directions. As shown in Fig. 5, two traverse paths ‘1→4→5→9’ and ‘1→6→5→9’ from node 1 to node 9 correspond to the same meta-path. The context-path density of this meta-path can be calculated as follows:

\[
p(1 \text{ to } 9) = \left(\frac{2}{5} \times \frac{4}{10}\right) \times \left(\frac{2}{5} \times \frac{4}{10}\right) \times \left(\frac{2}{5} \times \frac{3}{10}\right) \times \left(\frac{2}{5} \times \frac{3}{10}\right)
\]

\[
= 0.1248
\]

Since not every target node has connectivity with the source node, we need to assign zero value to such target nodes after the TSDW calculation of each snapshot. This is also done to facilitate the samples and features alignment of the embedding vector in each snapshot.

VI. EXPERIMENTS

We conduct a thorough assessment of the proposed TSDW algorithm. We first introduce the dataset used in the experiment and its preprocessing. Then, the results obtained under different parameters (size of the time window, and depth-first meta-path search length) are compared horizontally. In summary, the experiments are designed to answer the following two research questions:

Question 1: How TSDW is superior to other state-of-the-art methods? (Section V)

Question 2: How do the time window and traverse path length affect TSDW? (Section VI)

A. EXPERIMENTAL SETUP

1) DATASETS

To evaluating prolific authors in a particular domain, we use the Aminer\(^3\) dataset, which is a platform for academic collaboration in the field of computer science.

In this dataset, the paper nodes contain the publication timestamps. It contains author nodes (A), paper nodes (P), venue nodes (V), domain nodes (D), and publication years in paper nodes, where domain-paper, author-paper, and paper-venue are heterogeneous edges, while paper-paper is a homogeneous edge. The graphical structure of the Aminer dataset and its node ratios are shown in Fig. 6, where we can see that the ratios of the four types of nodes are extremely unbalanced which are common in the real world.

In our study, we extract data of 20 years. The extracted dataset contains 4,348 authors, 98 venues, and 113,927 papers. We choose the ‘Artificial Intelligence (AI)’ domain node as the source node and all the author nodes as the target nodes.

\(^3\)https://www.aminer.cn/
TABLE 2. The preprocessed datasets.

| Preprocessed Aminer dataset | Nodes | Links |
|-----------------------------|-------|-------|
| Author                      | 4348  |       |
| Paper                       | 113927| 98    |
| Venue                       |       | 1     |
| Domain                      |       |       |
| A-P                         | 66073 |       |
| D-P                         | 29115 |       |
| V-P                         | 108959|       |
| P-P                         | 301427|       |

searches for author nodes \( V_{\text{author}} \) in the domain node \( V_{\text{AI}} \).

The traversal path length \( L_{\text{max}} \) and time window size \( W_{\text{size}} \) are taken as the variable parameters.

2) METHODS COMPARED

To ensure equitability when comparing, all the ‘paper-domain’ links excluding those related to the ‘AI’ domain are deleted, given that the techniques to be compared are independent of any specific discipline domains. Nevertheless, it is concerned that some methods of comparison adopting the random walk strategy are easily compromised due to other domain type nodes. Sun et al. [4] pointed out that the paths generated in a random walk are biased to highly visible and more concentrated nodes.

The citation links between papers are not deleted as there are many papers in the dataset which have no ‘paper-domain’ links. The detailed descriptions of the dataset are shown in Table 2. After preprocessing the dataset, we compare our algorithm with the following state-of-the-art heterogeneous network embedding baselines.

- **DeepWalk** [16]. It learns node embedding by performing a classical random walk on a network, and then sends the resulting node sequences to the SkipGram model. DeepWalk adopts average sampling and assumes that the random walk node sequences resemble the context-interpreted words. We set the number of walks per node \( r = 10 \), the maximum walk length \( L_{\text{max}} = 128 \), and the window size for the SkipGram model \( W_{\text{window}} = 10 \).

- **LINE** [26]. Unlike the DFS used by DeepWalk, LINE can be considered as a method which adopts the BFS (Breadth-First Search) strategy. Two similarities are defined in LINE to further deepening node relationships, enabling trained embeddings to preserve both local and global network structures. It is a method based on the proximity assumption of the first and second order nodes. LINE learns the node similarities of the first and second orders separately and combines them to perform node embedding.

- **Node2vec** [15]. It is an embedding method that integrates the depths and breadths of nodes, which can be considered as an extension of DeepWalk. It integrates DFS and BFS random walk strategies. Besides, Node2vec uses the Alias algorithm [22] to sample adjacent nodes, after which Word2vec [25] is also adopted to learn the embedding vector of the nodes. We set the number of walks per node \( r = 10 \), the maximum walk length \( L_{\text{max}} = 128 \), and the window size for the SkipGram model \( W_{\text{window}} = 10 \).

- **PTE** [17]. It first assumes that the heterogeneous networks can be decomposed into several bipartite networks with only two types of nodes, and then summarizes the respective loss functions in the embedding and prediction tasks. Similar to [7], we use PTE in an unsupervised way. Specifically, we decompose the Aminer heterogeneous network into four types of bipartite networks as follows: author–paper (A-P), venue–paper (V-P), domain–paper (D-P), paper–paper (P-P), and then feed these four bipartite networks to PTE to output the node embedding and clustering.

- **Metapath2vec** [7]. Like DeepWalk, it attempts to build a network structure using a random walk. However, unlike in DeepWalk, the conditional probability, which determines the next step of the walk in a heterogeneous network, cannot be normalized directly over all the neighbors of the next node as it ignores the type information of the node. Therefore, to capture the semantic and structural correlations between different types of nodes, metapath2vec proposes the random walk based on meta-path. In other words, its random walk strategy is based on a pre-set meta-path condition, and then, the obtained node sequences are fed to a SkipGram model.

- **HIN2vec** [8]. It learns node embedding and meta-path representation jointly by combining a set of meta-paths shorter than a given length to guide the random walk. Keeping all other parameters the same as in DeepWalk, we perform experiments by varying the meta-path maximum length from 3 to 6 and report the best results.

- **CTDNE** [18] is also developed based on a random walk strategy to obtain a training corpus, which is fed to the Skip-Gram model to obtain network embeddings. The strategy creates events based on a chronological order, specifying that the timestamp of the following edge must be greater than that of the present edge in the walking process. Theoretically, the set of the random walk sequences in temporal order is a subset of the non-temporally ordered sequences set. According to the information theory, the inclusion of temporal information reduces embedding uncertainty and contributes to better performances than many other algorithms, such as DeepWalk and Node2vec, on traditional tasks. To ensure the equitability of the experiment, we set the same parameters as those in DeepWalk.

- **HTNE** [33] attempts to capture the variations in nodes and edges of a dynamic network in order to maintain the network structure during embedding. It models the evolution of nodes through their neighborhood formation sequence and then captures the influence of historical neighbors on the
Table of Contents

Q. Hu et al.: Task-Guided Context-Path Embedding in Temporal Heterogeneous Networks

FIGURE 7. Performance on multi-class node classification.

FIGURE 8. Performance on node clustering.

current neighborhood formation sequence using the Hawkes process. We set the parameters as provided by Zuo et al. [33]. Note that among the above baselines, DeepWalk and LINE are for homogeneous networks and they ignore the otherness between different nodes and links in a network. For all the methods, we focus on the embedding of authors in the ‘AI’ domain and evaluate their performances by multi-class node classification and clustering.

In our proposed method TSDW, we set the domain type node ‘AI’ as the source node and the target nodes are the set of author nodes. The size of the sliding time window is set as \( w_{size} = \{3, 4, 5\} \), which means that the time window takes 3 (or 4, or 5) years as the window size and slides in the 20-year interval. The maximum path length \( L_{\text{max}} \) is set in \( \{3,4,5\} \). For example, in Fig. 1(a), if \( L_{\text{max}} = 5 \), we can extract five different types of meta-paths (‘D-P-A’, ‘D-P-P-A’, ‘D-P-V-P-A’, ‘D-P-P-P-A’, ‘D-P-A-P-A’) in the set of context-paths. As we can see, there is no limit to the symmetry of meta-paths. By tuning the two parameters, we report our best performance and investigate the impact of these parameters in Section VI.

B. MULTI-CLASS NODE CLASSIFICATION

Multi-class classification means a classification task with more than two classes. It assumes that each sample is assigned to one and only one label. To accomplish the specific task of this paper (‘how to embed the output of scholars in a given domain?’), we focus on the production of authors, which means the number of papers published by the authors within the temporal frame. The number of papers published by authors in the ‘AI’ domain are employed as labels, so that this task can be implemented. To be more precise, the number of published papers more than (including) 20 is treated as the most prolific label, between 10 (including 10) and 20 as the more prolific label, and 0 to 10 as the general prolific label. We randomly select a group of author nodes as labeled nodes for the purpose of training and then train a linear SVM classifier to predict the most likely labels for test nodes and compare the prediction against their ground truth labels. We report the test size as well as the relevant Macro-F1 and Micro-F1 scores.

Fig. 7 shows our performance on the classification task. Firstly, we observe that more training data lead to better classification performance. Secondly, we find that our method is generally above the baselines, except HIN2vec which shows comparable performance with TSDW as it extracts critical meta-paths information, such as ‘D-P-A’ and ‘A-P-A’. However, compared to HIN2vec which combines all meta-paths shorter than the specified length, the context-paths embedded from the time series established by TSDW are more explanatory, and TSDW can store the network’s historical pieces of information offline.

C. NODE CLUSTERING TASK

Clustering refers to a method that automatically divides a bunch of data into several classes, which belong to the unsupervised learning method. This method should ensure that the data of the same class have similar characteristics. We also focus on the production of authors and regard nodes with the same label as a ground-truth community. We adopt the \( k \)-means [27] algorithm to cluster embedding vectors of nodes and evaluate the clusters using the Normalized Mutual Information (NMI).

Fig. 8 shows our node clustering task, where we observe that the clustering performance of our proposed TSDW is significantly better than those of all other baselines. However, further observation shows that the labels of author nodes tend to degrade the clustering performance as the labels (quantities of papers) are a kind of relationship of inclusion, specifically for the TSDW algorithm. The embedding vectors obtained from the context-path density between different labels of nodes have similar eigenvectors from the high-dimensional perspective.

Besides, the meta-path based random walk strategy is inadequate in the networks with severely unbalanced types of nodes. In addition, insufficient core nodes can lead to the
TABLE 3. Impact of the window size $w_{\text{size}}$ and maximum path length $L_{\text{max}}$ in $[3,5]$.

| Dataset | Time window $w_{\text{size}}$, Path length $L_{\text{max}}$ | Classification 20% | Classification 40% | Classification 60% | Classification 80% | Clustering NMI |
|---------|----------------------------------------------------------|---------------------|---------------------|---------------------|---------------------|----------------|
|         |                                                          |                     |                     |                     |                     |                |
|         | $w_{\text{size}} = 3, L_{\text{max}} = 3$              | 74.99 / 85.54       | 76.54 / 85.45       | 78.49 / 86.97       | 79.27 / 86.53       | 0.4487         |
|         | $w_{\text{size}} = 3, L_{\text{max}} = 4$              | 80.06 / 88.00       | 80.70 / 88.43       | 84.46 / 90.43       | 85.49 / 90.53       | 0.4805         |
|         | $w_{\text{size}} = 3, L_{\text{max}} = 5$              | 75.81 / 87.16       | 82.21 / 90.02       | 84.73 / 90.77       | 86.39 / 92.34       | 0.4031         |
| Amine   | $w_{\text{size}} = 4, L_{\text{max}} = 3$              | 75.00 / 85.54       | 76.90 / 85.53       | 78.68 / 87.20       | 79.38 / 86.58       | 0.4487         |
|         | $w_{\text{size}} = 4, L_{\text{max}} = 4$              | 80.09 / 88.03       | 80.71 / 88.43       | 84.17 / 90.20       | 85.49 / 90.64       | 0.4855         |
|         | $w_{\text{size}} = 4, L_{\text{max}} = 5$              | 76.18 / 87.39       | 82.52 / 89.57       | 86.36 / 91.10       | 88.36 / 92.12       | 0.3720         |
|         | $w_{\text{size}} = 5, L_{\text{max}} = 3$              | 75.18 / 85.59       | 76.68 / 86.45       | 78.78 / 87.25       | 78.58 / 86.60       | 0.4487         |
|         | $w_{\text{size}} = 5, L_{\text{max}} = 4$              | 80.10 / 88.06       | 80.74 / 88.43       | 84.37 / 90.20       | 85.51 / 91.54       | 0.4855         |
|         | $w_{\text{size}} = 5, L_{\text{max}} = 5$              | 74.99 / 86.71       | **82.65** / 89.34   | **86.42** / 91.65   | **88.32** / 92.57   | 0.3770         |

Algorithm distracted by numerous other types of nodes (the paper type nodes in this study). Therefore, it is not easy for the algorithm to effectively extract sufficient meta-paths, such as Metapath2vec.

That is to say, the random walk tends to the nodes of the domain of high visibility, which is easy to be biased by a large number of suboptimal meta-paths [24].

VI. IMPACT OF THE WINDOW SIZE $w_{\text{size}}$ AND MAXIMUM PATH LENGTH $L_{\text{max}}$

We investigated the impact of the window size $w_{\text{size}}$ and maximum path length $L_{\text{max}}$. The size $w_{\text{size}}$ balances the length of time (year) span when the time window slides. A larger span size enables more nodes and links in the snapshot, whereas a smaller $w_{\text{size}}$ enables the meta-path search algorithm to perform more efficiently in the snapshot. The longer the path length, the more meta-path types can be extracted. For example, when $L_{\text{max}} = 5$, five different types of meta-paths between the source node and the target node can be extracted. Note that semantic and structural connections between two nodes are directly proportional to the types of meta-paths.

Table 3 shows how the results were affected by the parameters. We observe that with the increasing size of the time window over the same path length, the improvement of the clustering performance is limited to the classification performance. That is, the primary positive effect of the time window size is to preserve historical information and reduce the complexity of the DFS algorithm. For many real-world datasets, historical nodes and relationships between them are essentially immovable. So, we can attempt to use the time window to extract and store historical information offline or in parallel.

From another point of view, in the case of a fixed time window, a longer path length can significantly improve the classification effect. The longer path length leads to a greater number of meta-path types in the context-path set obtained by the TSDW algorithm. For example, the ‘D-P-A’ meta-path contains the most abundant semantic, and the longer meta-paths such as ‘D-P-V-P-A’ reflects a better structure detail in the network. As shown in Table 3, under the consideration that the complexity of the algorithm can be tolerated, a larger time window and a longer path length can indeed improve the classification performance. Therefore, the meta-paths of various lengths and types can reflect the attributes of some aspects of the network, and lead to feature enhancement of samples.

Another interesting phenomenon is that a path length equal to 5 could achieve the optimal classification performance, but the optimal clustering performance when the path length equals 4. We further observe that the meta-path types are ‘D-P-A’ and ‘D-P-P-A’ for $L_{\text{max}} = 4$, which have rich semantics; whereas when $L_{\text{max}} = 5$, the meta-path types also include ‘D-P-A-P-A’, ‘D-P-V-P-A’, and ‘D-P-P-P-A’, which focus more on structural information. That is, different meta-paths have different effects on information expression, thus leading to conflicts. Excessively long suboptimal meta-paths bring extra noise in the clustering task.

What we want to emphasize in this study is that the time window should not be less than 3. We observe that when the time window is 2, the network structure becomes very sparse and many empty context paths emerge between nodes, resulting in a large number of zero in the node embedding vectors. On the other hand, as mentioned above, an increasing size of the time window does not significantly improve the performance. So the path length cannot be set to less than 3 or longer than 5. When it is less than 3, there exists no context path connection between the source node and the target nodes. And when it is longer than 5, there exist too many context paths. This is unacceptable in terms of the complexity of the TSDW algorithm.

VII. DISCUSSION

A. COMPUTATIONAL COMPLEXITY ANALYSIS

Let $\mathcal{V}$ be the set of network nodes, and $\mathcal{E}$ be the set of network edges. Methods based on the random walk [7], [15], [16] hold a time complexity of $O(|\mathcal{V}|^2)$. These methods generate
\[ |V| r \text{ walks each of length } l_{\text{max}}, \text{ and then calculate the final embedding of } |V| d \text{ (d is the embedding size in the Skip-Gram model parameters). The lowest time complexity among other state-of-the-art temporal methods is } O(|V|^2) \] 

Let the average number of all the context-path sets \( N \) between two nodes be \( \psi \). Our algorithm uses a modified depth-first search [32] to generate the context-paths, where a single context-path can be found in \( O(|V| + \epsilon) \). For all the nodes in the network, the complexity of our TSDW algorithm is \( O(\psi |V| |V| + \epsilon) \). Since most of the real networks are sparse, \(^4\) the complexity can be abbreviated as \( O(|V| |V| + \epsilon) \). For the TSDW algorithm, we assume that the historical interaction information is immutable and the snapshots obtained by using temporal segmentation can be saved offline, which means that the number of nodes and edges in a new snapshot is much smaller than that in the original network. We use a parallel processing approach to keep the complexity of the algorithm within an acceptable range.

**B. TEMPORAL HETEROGENEOUS NETWORK AT THE MILLISECOND LEVEL**

In an academic network (Aminer, DBLP, etc.), heterogeneous networks can be divided based on the year stamp to reduce complexity and establish a time sequence, enabling the contextual relationship between nodes. However, in many heterogeneous networks, the timestamps in the nodes or links maybe in minutes or even in milliseconds (e.g., e-commerce networks and disaster warning networks). In these types of networks, the TSDW strategy will be inadequate if timing segmentation and time window sliding are used. Embedding on heterogeneous networks with rich and tiny time granularity is still a huge challenge to network data-mining research.

**C. EFFECTIVE COMBINATION OF RICH SEMANTIC META-PATHS AND RICH STRUCTURAL META-PATHS**

In the past, network embedding was usually limited to homogeneous networks in studying the structure information of networks. The introduction of heterogeneous networks has greatly contributed to the progress of network mining research, mainly to grasp semantic relationships. In our study, we attempt to combine semantic and structural information. However, instead of being directly implemented by weighting or normalizing, another approach to combine them is required. Synchronously, the development of neural networks and deep learning has greatly helped to improve the performance of heterogeneous network embedding and prediction. The interpretability of the semantic existence in heterogeneous networks has also been diluted. It is still a meaningful attempt to effectively combine semantics and structure in heterogeneous networks.

**VIII. CONCLUSION**

A network is the best data structure for representing real-world interactions. Heterogeneous networks greatly improve the semantic expression of a network. Existing heterogeneous network embedding methods are based on the random walk (or meta-path guided random walk), which extracts the structure and semantics of a network through the walk sequence of nodes. However, it is difficult to maintain a balance between semantics and structure. In this paper, a snapshot sequence based on the timestamp is established following the process of starting with a specific task in a heterogeneous network, and then segmenting the timestamps and adopting the strategy of time window sliding. Then, a context-path vector between two nodes is computed through the depth-first meta-path search algorithm. Finally, all the obtained context-path vectors are merged and normalized in the scale of the whole heterogeneous network to realize the embedding between the two nodes based on a time sequence. More precisely, we propose TSDW, a task-guided context-path embedding in temporal heterogeneous networks, which not only utilizes the time information but also effectively balances the semantics and structure. To validate this technique, we evaluate the multi-class node classifications. Also, we perform clustering on the Aminer dataset and analyze the influence of the TSDW related parameters on the performance. Empirical results for network data show that TSDW could significantly outperform the state-of-the-art representation learning models, including DeepWalk, LINE, Node2vec, PTE, Meapath2vec, HIN2vec, HTNE, and CTDNE by 3.02% to 44.9% of Macro-F1, 0.9% to 18.92% of Micro-F1 in multi-class node classification and 21% to 47% of NMI in node clustering.

In our future work, an extension of our framework will be intended to ensure a more effective balance between structure and semantics. Also, the strategy will be applied to smaller granular time information in temporal heterogeneous networks.

**REFERENCES**

[1] H. Cai, V. W. Zheng, and K. C.-C. Chang, “A comprehensive survey of graph embedding: Problems, techniques, and applications,” IEEE Trans. Knowl. Data Eng., vol. 30, no. 9, pp. 1616–1637, Sep. 2018, doi: 10.1109/TKDE.2018.2807452.

[2] Y. Li, Z. Lou, Y. Shi, and J. Han, “Temporal motifs in heterogeneous information networks,” in Proc. MLG Workshop@ KDD, 2018, doi: 10.475/123.

[3] J. D. Noh and H. Kieger, “Random walks on complex networks,” Phys. Rev. Lett., vol. 92, no. 11, Mar. 2004, Art. no. 118701, doi: 10.1103/PhysRevLett.92.118701.

[4] 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, Aug. 2011, doi: 10.14778/3402707.3402736.

[5] 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, New York, NY, USA: Association for Computing Machinery, 2017, pp. 295–304, doi: 10.1145/3018661.3018735.

[6] M. J. J. Han, and M. Danilevsky, “Ranking-based classification of heterogeneous information networks,” in Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, New York, NY, USA: Association for Computing Machinery, 2011, pp. 1298–1306, doi: 10.1145/2020408.2020603.

[7] Y. Dong, N. V. Chawla, and A. Swami, “Metapath2vec: Scalable representation learning for heterogeneous networks,” in Proc. 25rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, New York, NY, USA: Association for Computing Machinery, 2017, pp. 135–144, doi: 10.1145/3097983.3098036.
[8] T. Mikolov, W.-C. Lee, and Z. Lei, “HIN2Vec: Explore meta-paths in heterogeneous information networks for representation learning,” in Proc. ACM Conf. Inf. Knowl. Manage. New York, NY, USA: Association for Computing Machinery, 2017, pp. 1797–1806, doi: 10.1145/3132847.3132953.

[9] X. Li, B. Kao, Z. Ren, and D. Yin, “Spectral clustering in heterogeneous information networks,” Proc. AAAI Conf. Artif. Intell., vol. 33, Jul. 2019, pp. 4221–4228, doi: 10.1609/aaai.v33i01.33014221.

[10] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu, “SHINE: Signed heterogeneous information network embedding for sentiment link prediction,” in Proc. 11th ACM Int. Conf. Web Search Data Mining, 2017, pp. 592–600, doi: 10.1145/3016024.3016039.

[11] Z. Huang and N. Mammoulis, “Heterogeneous information network embedding for meta path based proximity,” 2017, arXiv:1701.05291. [Online]. Available: http://arxiv.org/abs/1701.05291.

[12] G. Kossinets, “Empirical analysis of an evolving social network,” Science, vol. 311, no. 5757, pp. 88–90, Jan. 2006, doi: 10.1126/science.1116869.

[13] X. Zhang, Z. Zhao, C. Li, Y. Zhang, and J. Zhao, “An interpretable and scalable recommendation method based on network embedding,” IEEE Access, vol. 7, pp. 9384–9394, 2019.

[14] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” CoRR. 2013.

[15] 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: Association for Computing Machinery, 2016, pp. 855–864, doi: 10.1145/2939672.2939754.

[16] 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: Association for Computing Machinery, 2014, pp. 701–710, doi: 10.1145/2623330.2623732.

[17] 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, New York, NY, USA: Association for Computing Machinery, 2015, pp. 1165–1174, doi: 10.1145/2783258.2783307.

[18] G. H. Nguyen, J. B. Lee, R. A. Rossi, N. K. Ahmed, E. Koh, and S. Kim, “Continuous-time dynamic network embeddings,” in Proc. Web Conf. Int. World Wide Web Conf. Steering Committee, Republic and Canton of Geneva, Switzerland, 2018, pp. 969–976, doi: 10.1145/3184558.3191526.

[19] T. K. Saha, T. Williams, M. Al Hasan, S. Joty, and N. K. Varberg, “Models for capturing temporal smoothness in evolving networks for learning latent representation of nodes,” 2018, arXiv:1804.05816. [Online]. Available: http://arxiv.org/abs/1804.05816.

[20] M. Rahman, T. K. Saha, M. Al Hasan, K. S. Xu, and C. K. Reddy, “DyLink2Vec: Effective feature representation for link prediction in dynamic networks,” 2018, arXiv:1804.05755. [Online]. Available: https://arxiv.org/abs/1804.05755.

[21] 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, doi: 10.1109/TKDE.2018.2833443.

[22] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola, “Reducing the sampling complexity of topic models,” in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, New York, NY, USA: Association for Computing Machinery, 2014, pp. 891–900, doi: 10.1145/2623330.2623756.

[23] C. Yang, Y. Xiao, Y. Zhang, Y. Sun, and J. Han, “Heterogeneous network representation learning: Survey, benchmark, evaluation, and beyond,” 2020, arXiv:2004.00216. [Online]. Available: https://arxiv.org/abs/2004.00216.

[24] R. Hessein, D. Yang, and P. Cudré-Mauroux, “Are meta-paths necessary? Revisiting heterogeneous graph embeddings,” in Proc. 27th ACM Int. Conf. Inf. Knowl. Manage, New York, NY, USA: Association for Computing Machinery, 2018, pp. 437–446, doi: 10.1145/3269206.3271777.

[25] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in Proc. 26th Int. Conf. Neural Inf. Process. Syst., vol. 2. Red Hook, NY, USA: Curran Associates, Inc., pp. 3111–3119.

[26] 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, Republic and Canton of Geneva, Switzerland, 2015, pp. 1067–1077, doi: 10.1145/2736277.2741093.

[27] D. Arthur and S. Vassilvitskii, “K-Means++: The advantages of careful seeding,” in Proc. 18th Annu. ACM-SIAM Symp. Discret. Algorithms, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.