Front.Comput.Sci.
DOI

SUPPLEMENTARY FILE

THS-GWNN: A Deep Learning Framework for Temporal Network Link Prediction (Supplementary File)

Xian Mo [1], **Jun Pang** [2], **Zhiming Liu** (✉)[1]

1 College of Computer and Information Science, Southwest University, Chongqing 400700, China
2 Faculty of Science, Technology and Medicine & Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Esch-sur-Alzette L-4364, Luxembourg

Abstract Link prediction in temporal networks whose edges evolve over time aims at inferring new edges based on a sequence of previous network snapshots. Network embedding is an important analytical tool for temporal network link prediction, which helps us better understand network evolution. How to encode high-dimensional and non-Euclidean network information is a crucial problem for node embedding in temporal networks. One of the challenges is to reveal the spatial structure at each timestamp and the temporal property over time. In this paper, we propose a graph wavelet neural network (THS-GWNN) framework, base on timestamp hierarchical sampling, for link prediction in temporal networks. More precisely, we develop a timestamp hierarchical sampling algorithm (THS) to capture spatial-temporal features, which samples the vertices from the current timestamp to the previous one and can well preserve the evolving behavior of temporal networks. Next, we adopt graph wavelet neural networks (GWNN) to embed the vertices and long-short term memory networks (LSTMs) for predicting new links. Extensive experiments on several datasets demonstrate that THS-GWNN can effectively predict links on temporal networks and it outperforms the state-of-the-art models.

Keywords Temporal networks, link prediction, network embedding, spatial-temporal features, wavelet neural networks, long short-term memory networks

Received month dd, yyyy; accepted month dd, yyyy

E-mail: *zhimingliu88@swu.edu.cn*

1 Introduction

Link prediction for temporal networks aims to evaluate the likelihood of the future linkage among nodes, which has significant applications in social networks [1,29], biological networks [2], traffic analysis [3], etc. It is also an important analytical tool for temporal networks, which helps us better understand network evolution [4]. For instance, we can predict which links will be established in the near future to predict new relationship in online social networks [1].

Many methods for link prediction in networks have been proposed in the literature, including [2, 5–11, 25]. Common Neighbours (CN) [27] and Resource Allocation Index (RA) [25] , are widely used in link prediction of static networks [4]. Both, however, depend on simple statistics of networks and thus can hardly deal with the evolving network structure directly. In recent years, network embedding techniques were proposed to learn the representations of networks, such as DeepWalk [16], node2vec [17], SDNE [18] and GCN [15]. These embedding methods are powerful but cannot still analyzing the evolution of networks. The spatial structure features and temporal evolution features of temporal networks are the key information for effective link prediction in temporal networks. The spatial structure features represent the topology relationship of the network, while the temporal evolution features represent networks topology evolving behavior from the current snapshot to the previous snapshot. For comprehend complex behaviors of temporal networks, it is essential to use both spatial and temporal features to reveal the spatial structure at each timestamp and the temporal property over time. One common approach utilizes various topological similarities [11], which predicts future similarity scores between nodes based on their past similarity score values. However, it captures segmentary features, but cannot capture underlying features of temporal networks. There exist another approach, based on non-negative matrix decomposition, is to explore networks' topology structure [5,7–10]. However, due to sparse and large-scale in real-life networks, methods using matrix decomposition may have high computational cost. In addition, the methods have limited ability to extract the correlation of high dimensional features [12]. Encoding high-dimensional and non-Euclidean network information is a challenging problem for learning node representations in temporal networks. However, the emergence of deep learning technology brings new insights for further research in this field. Li et al. [6] develop a DDNE model to capture both spatial and temporal features using GRUs. Chen et al. [24] develop an end-to-end E-LSTM-D model to integrate a stacked LSTM into the architecture of encoder-decoder. However, the input to these model are the adjacent matrix of the networks, hence this also causes high computational cost. tNodeEmbed [31] learns the evolution of a temporal network's nodes and edges over time and incorporates dynamics in a temporal node embedding framework, while DCRNN [32] proposes a diffusion convolutional recurrent neural network to captures the spatio-temporal dependencies. To achieve effective traffic prediction, STGCN [29] replaces regular convolutional and recurrent units that integrating graph convolution and gated temporal convolution and T-GCN [30] combines with the graph convolutional network (GCN) and the gated recurrent unit (GRU) to capture spatial-temporal features. The flexible deep embedding approach (NetWalk) [23] utilises an improved random walk to extract the topological and temporal features of the network. More recently, DySAT [21] computes node representations through joint self-attention along with the two dimensions of the structural neighborhood and temporal dynamics, and dyngraph2vec [8] learns the temporal transitions in the network using a deep architecture composed of dense and recurrent layers. However, these methods are not considered combining the previous snapshot to extract spatial-temporal features for each node of the current snapshot with weight. Therefore, the representation ability of temporal networks is still insufficient.

To tackle the above identified problems, we propose a novel model named THS-GWNN for link prediction in temporal networks, an overview of our THS-GWNN model is described in Figure 1 (a: Raw input which is a temporal network G_t; b: A spatial-temporal feature extraction layer which extracts $\Gamma(v, k, t_{Sta}, t_{End})$ of each vertex v of each network snapshot; c: An embedding layer which maps each vertex to its D-dimensional representation; d: A LSTM layer which builds the model for link prediction; e: Model output.). The model adopts efficient neural networks to deeply embed spatial and temporal features, which can effectively predict links for temporal networks. In temporal networks, the current snapshot topology is derived from the previous snapshot topology, so we combine the previous snapshot to extract spatial-temporal features for each node of the current snapshot. Inspired by this idea, we propose a timestamp hierarchical sampling algorithm (THS) to capture both spatial and temporal features of the networks, which samples neighbours of a given node from the current network snapshot to the previous snapshot. Because nodes with a fewer hops to the current node normally have closer relations to the node and snapshots closer to the current snapshot have closer relations to the current snapshot, we add a decaying exponential to ensure that the fewer hops and the closer snapshots, the more nodes are sampled. In this way, THS can better preserve both spatial structure and temporal evolution features of the networks. Besides, it is a method based on local feature extraction, which can decrease input features dimension and improve efficiency. We then adopt graph wavelet neural networks (GWNN) [13] to embed the spatial-temporal features into vectors. During the link prediction phase, we use long short-term memory networks (LSTMs) [14] to capture the time dependence among network snapshots.

Our major contributions in this work can be summarised as follows.

- We propose a model THS-GWNN to perform link pre-

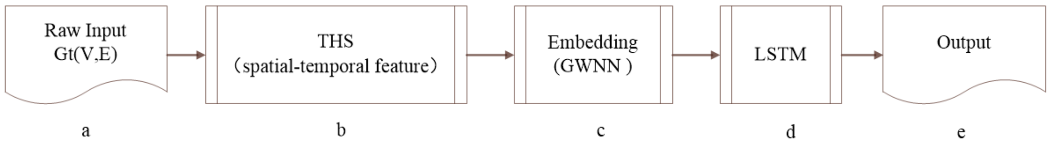

Fig. 1 Overview of the framework THS-GWNN.

diction in temporal networks. The model adopts a graph wavelet neural network (GWNN) to deeply embed nodes, which can better capture the nonlinear features of temporal networks.

- We propose a timestamp hierarchical sampling algorithm (THS) for both spatial and temporal feature extraction, which can effectively capture the evolving behavior of temporal networks. It samples neighbors for the current node v from the current snapshot' K-hop neighbours to the previous snapshots' K-hop neighbours, which can hierarchically extract both spatial and temporal features for the nodes. It also incorporates a decaying exponential to assign the more sampled nodes to the fewer hops and the closer snapshots, which can better preserve the evolving behavior of temporal networks.
- Experiments on four real-world datasets (i.e., Facebook friendships, Hep-Ph, Digg and Facebook wall posts) demonstrate that our THS-GWNN outperforms a few state-of-the-art baseline models.

The rest of the paper is arranged as follows. Section 2 summarises several related works. We formulate the problem and summarise the notations in Section 3. Sections 4 presents in detail our proposed framework THS-GWNN for link prediction in temporal networks. Section 5 discuss the experimental results, and we conclude the paper in Section 6.

2 Related Work

In this section, we briefly summarise related work for link prediction in temporal networks. A recent survey [4] indicates that link prediction in static networks has been extensively studied. Due to the fact that networks are continually evolving with time in the real world, so it is necessary to study link prediction in temporal networks. One common approach utilizes various topological similarities, such as common neighbors [11], T-Flow [26], HPLP [20] etc. The [11] define a time series model to predict future similarity scores value between nodes. T-Flow [26] consider link activeness to computes information flow between nodes. HPLP [20] combines various topological information (node degree and common link predictors) into a bagged random forests classification framework to supervised predict link. However, these methods capture segmentary features, and cannot capture the underlying features of temporal networks. There exists another approach based on matrix decomposition to explore the spatial topology of the networks [7–9]. The main idea of the method is that the closer two nodes of current snapshot are, the more likely they are to form a link in the near future snapshot. However, real-life networks are often evolving, methods considering only spatial information may have inefficient performance for predicting link. There exist a few other methods focusing on both spatial and temporal evolution features, such as SETP [5], LIST [10]. The SETP [5] constructs a sequence of higher-order proximity matrices to capture the implicit relationships among nodes, while the LIST [10] defines the network dynamics as a function of time, which integrates the spatial topology of networks at each timestamp and the temporal network evolution. Because they are still based on matrix decomposition, both of them have limited ability to extract the correlation of high dimensional features.

In recent years, a network embedding approach based on neural networks has gained a lot of popularity [16], and it aims to map nonlinearly each node of the network into a low-dimensional space. This approach has proved to be very effective in temporal networks for link prediction [6]. Many different network embedding methods using neural networks have been proposed, including DeepWalk [16], node2vec [17], and SDNE [18]. However, these methods focused on representation learning for static networks and cannot obtain temporal features of temporal networks. The above discussed models principally focus on shallow models with limited ability to extract nonlinear features. Thus, it is necessary to extend neural network-based methods for aggregating both spatial and temporal feathers using deep model for link prediction. Li et al. [6] develop a link prediction model capturing both spatial and temporal features using GRUs inspired by the machine translation problem of encoder-decoders methods [19]. Chen et al. [24] develop an end-to-end E-LSTM-D model to integrate a stacked LSTM into the architecture of encoder-decoder. It imposes more penalty to exist links in the objective to cope with the problem of sparsity. However, the input to these model is the adjacent matrix of the networks, hence it has high computational cost. The T-GCN [30] combines with the graph convolutional network (GCN) to capture spatial dependence and the gated recurrent unit (GRU) to capture temporal dependence. tNodeEmbed [31] presents a joint loss function that creates a temporal embedding of a node by learning to combine its historical temporal embeddings to learns the evolution of a temporal network's nodes and edges over time. DCRNN [32] adopts the encoder-decoder architecture, which uses a bidirectional graph random walk to model spatial dependency and recurrent neural network to capture the temporal dependencies. STGCN [29] integrates graph convolution and gated temporal convolution through Spatio-temporal convolutional blocks to capture spatio-temporal features. The DySAT model [21] stacks temporal attention layers on top of structural attention layers to learn node representations, which computes node representations through joint self-attention along with the two dimensions of the structural neighborhood and temporal dynamics, while dyngraph2vec [8] learns the temporal transitions in the network using a deep architecture composed of dense and recurrent layers, which learns the structure of evolution in dynamic graphs and can predict unseen links. NetWalk [23] is an flexible deep embedding approach, and it uses an improved random walk to extract the topological and temporal features of the network. The approach can update the network representation dynamically as the network evolves by clique embedding. However, these methods are not considered combining the previous snapshot to extract spatial-temporal features for each node of the current snapshot with weight, where (1) nodes with smaller hops from the current node contribute more to the current node for spatial features and (2) snapshots closer to the current snapshot contribute more to the current snapshot for temporal features. Therefore, the representation ability of these methods for temporal networks is still insufficient.

3 Problem Definition

In this section, we introduce some definitions and formally present our research problem in this paper.

Definition 1 [Network] A network can be represented graphically: $G = \langle V, E\rangle$, where $V = \{v_1, \ldots, v_n\}$ represents a set of nodes, and n is the number of nodes, and $E \subseteq V \times V$ represents a set of links (edges) among nodes.

Definition 2 [Temporal network] We follow the temporal network settings in [6] that the set of nodes is fixed, while the edges E_t can evolve over time. Hence, a temporal network is defined as $G_t = \langle V, E_t\rangle$, which represents a network $G = \langle V, E\rangle$ evolving over time and generates a sequence of snapshots $\{G_1, \ldots, G_T\}$, where $t \in \{1, \ldots, T\}$ represents the timestamps.

Definition 3 [Node K-hop neighbours] Let $G_t = \langle V, E_t\rangle$ be a temporal network. For a node v in timestamp t, its K-hop neighbours can be defined as the set $N^t(v, K) = \bigcup_{k=1}^{k=K}\{R^k(v, {A_t}^k)\}$, which contains some of the the neighbours of v with K hops in the timestamp t. The value k represents the hop, and $R^k(v, {A_t}^k)$ represents the set of randomly sampled neighbors of v in hop k in timestamp t and ${A_t}^k$ represents the number of neighbors sampled of v in hop k in timestamp t and $|R^k(v, {A_t}^k)| \leq {A_t}^k$, where $|R^k(v, {A_t}^k)|$ represents the number of neighbors sampled of $R^k(v, {A_t}^k)$. We set $\widehat{A_t}^{k-1} = \gamma {A_t}^k$ and $\widehat{A}^1_{t-1} = \gamma {A_t}^1$ (t>1), where γ is a decaying exponential between 0 and 1 and $\widehat{A}$ represents rounding up A.

Definition 4 Let $G_t = \langle V, E_t\rangle$ be a temporal network. For a node v, its all K-hop neighbours from time t_{Sta} up to time t_{End} (i.e., $t_{Sta} \leq t_{End}$) are defined as $\Gamma(v, K, t_{Sta}, t_{End}) = \bigcup_{t=t_{Sta}}^{t_{End}}\{N^t(v, K)\}$, where t represents the timestamps, and K represents the number of the hops, and $N^t(v, K)$ is a set of K-hop neighbours of v in a network snapshot G_t where $t \in \{t_{Sta}, \ldots, t_{End}\}$.

Suppose we set $A^1_2 = 100$, $k = 3$, $\gamma = 0.8$, t_{Sta}=1 and t_{End}=2. According to Definitions 3 and 4, $A^2_2 = \widehat{A^2_2} = 0.8 * A^1_2 = 80$, $A^3_2 = \widehat{A^3_2} = 0.8 * A^2_2 = 64$, $A^1_1 = \widehat{A^1_1} = 0.8 * A^1_2 = 80$, $A^2_1 = \widehat{A^2_1} = 0.8 * A^1_1 = 64$ and $A^3_1 = \widehat{A^3_1} = 0.8 * A^2_1 = 51$. For a node v of timestamp t_{End}, it represents that we sample the number of neighbors with 1-hop less than 100 and 2-hop less than 80 and 3-hop less than 64 in current snapshots and 1-hop less than 80 and 2-hop less than 64 and 3-hop less than 51 in previous snapshots.

Link prediction for temporal networks. For a temporal network G_t that generates a sequence of snapshots $\{G_1, \ldots, G_T\}$, we respectively use to adjacency matrix $\{A_1, \ldots, A_T\}$ to describe its static topological structure. For the A_t (t represents timestamp), it is a 2-D array that stores the vertex relationships. The element in the A_t can be represented as a_{ij}, where i and j represent rows and columns of the 2-D array, respectively. If $a_{ij} = 0$, there is no edge between vertex i and j, otherwise, there is an edge. Temporal link prediction aims at predicting the adjacency matrix A_{T+1} at timestamp $T + 1$ according to the previously adjacency matrix $\{A_1, \ldots, A_T\}$.

4 THS-GWNN

In this section, we introduce our model, THS-GWNN, for temporal network link prediction, which is shown in Figure 1. In our model, we first propose a timestamp hierarchical sampling (THS) algorithm to extract spatial-temporal features. The extracted spatial-temporal features are then fed into a Graph Wavelet Neural Network (GWNN) for network embedding . Finally, LSTMs are adapted to predict new links.

4.1 Spatial-temporal feature extraction

For a given temporal network $G_t = \langle V, E_t \rangle$, the classical methods of sampling neighbor node use DeepWalk [16] and node2vec [17], which only capture spatial structure features and ignore temporal evolution features. Instead, we propose the THS algorithm to sampling $\Gamma(v, K, t_{Sta}, t_{End})$ for each node v. It samples neighbours for the current node v from the current snapshot' K-hop neighbours to the previous snapshots' K-hop neighbours, in this way THS can extract both spatial and temporal features for each node. Because nodes with smaller hops from the current node and snapshots with closer snapshot from the current snapshot potentially contribute more to the current node, we add a decaying exponential γ to ensure that the fewer hops and closer snapshots, the more nodes are sampled (see its details in Definition 3 and Definition 4).

The THS Algorithm has four input parameters: A_t^1: the number of sampled neighbors of v in 1-hop at timestamp t, K: the number of hops defining the neighbours of distance from one to at most K to a given node v, L: a window size defining how many previous network snapshots are taken into account when sampling v's neighbours and γ: the decaying exponential defining the fewer hops of the current snapshot to have more sampled nodes. We define $X[i]$ to represent the sampled neighbours for all nodes in the network at timestamp i, where $X[i]$ with $i \in \{1, \ldots, T\}$. The THS extracts spatial-temporal features from L snapshots to better simulate the evolutionary behavior of the temporal network. If the number of the previous snapshots is greater than N, sampling neighbour nodes is performed between $i - L + 1$ and i snapshots. Otherwise, it is only sampled from the very first snapshot 1 to the current snapshot i.

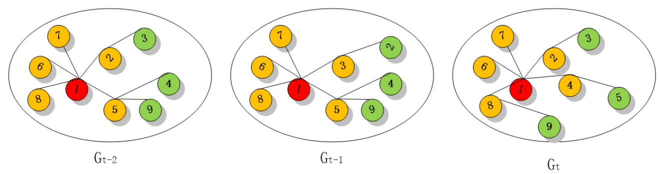

Fig. 2 The THS algorithm: an illustrative example

As shown in Figure 2, there are 9 nodes in the temporal network, and we extract features with the previous 2 snapshots for the node 1 in the current snapshot G_t. The orange nodes are the 1-hop neighbor of node 1, and the green nodes is the 2-hop neighbor of node 1. If we set A_t^1=4, K=2, L=2 and γ=0.8, then for the snapshot G_t the number of 1-hop sampled nodes A_t^1 is 4 and the number of 2-hop sampled nodes A_t^2 is 3. Hence, the multi-set of sampled nodes is {2,4,7,8,3,5,9}. For the snapshot G_{t-1}, the number of 1-hop sampled nodes A_{t-1}^1 is 3 and the number of 2-hop sampled nodes A_{t-1}^2 is 2. the multi-set of sampled nodes is {3,5,7,8,4,9}, and for the snapshot G_{t-2}, the number of 1-hop sampled nodes A_{t-2}^1 is 2 and the number of 2-hop sampled nodes A_{t-2}^2 is 2, the multi-set of sampled nodes is {2,5,3,4}. Then we combine the previous 2 snapshots sampled nodes as the final features for the node 1 of the snapshot G_t: {2,4,7,8,3,5,9,3,5,7,8,4,9,2,5,3,4}.

4.2 Neural network model

Our neural network model consists of the embedding layer and the LSTM layer as described in Figure 1.

4.2.1 Embedding layer

Network embedding [16] aims to map the network spatial-temporal properties into a low-dimensional matrix $X \in R^{D\times|V|}$, where each column represents the representation of a node in the network. In the model, we adopt Graph Wavelet Neural Network (GWNN) to map nonlinearly a node v to its $D-$dimensional representation $x_v \in R^D$. The GWNN leverages, base on graph convolution neural network architecture, wavelet transformer instead of the Fourier transformer of GCN. It does not require the eigendecomposition of the Laplacian matrix and thus is more efficient than traditional Graph Convolutional Networks (GCN) [15]. (see [13] for the time complexity analysis of the GWNN).

The GWNN was originally applied to static networks, and we design a m-layer GWNN for each snapshot for unsupervised node learning for temporal networks embedding. For the m-th layer GWNN, the input to each GWNN layer is a node feature matrix, X^1, with dimensions $n \times p$ and the output tensor is X^{m+1} with dimensions $n \times c$. The formulation of our model for each snapshot is (the framework of GWNN is described in Figure 3)

$$X^2_{[:,j]} = ReLU(\psi_s \sum_{i=1}^{p} F^1_{i,j}\psi^{-1}X^1_{[:,i]}) \quad j = 1, \ldots, q \qquad (1)$$

$$\vdots$$

$$X^{m+1}_{[:,k]} = ReLU(\psi_s \sum_{i=1}^{q} F^m_{i,j}\psi^{-1}X^m_{[:,i]})\ k = 1, \ldots, c \qquad (2)$$

where $X^1_{[:,i]}$ with the dimensions $n \times 1$ is the i-th column of X^1, $ReLU$ is a non-linear activation function, ψ_s is wavelet bases, ψ^{-1} is the graph wavelet transform matrix at scale s which projects signal in vertex domain into spectral domain, $F^n_{i,j}$ is a diagonal filter matrix learned in spectral domain in layer n [13], c is the embed dimension of each node, X^{m+1} of dimensions $n \times c$ is the embedding matrix of networks. We define the following loss function for each snapshot embedding to train the model:

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(X^{m+1}_{[:,i]} - Average(Adj(X^{m+1}_{[:,i]})))^2 \qquad (3)$$

where n is the number of nodes, and $X^{m+1}_{[:,i]}$ represents the vector representation of the node i in layer $m+1$,and $Adj(X^{m+1}_{[:,i]})$ obtains neighborhood node representation of i. $Average$ means an average processing operation.

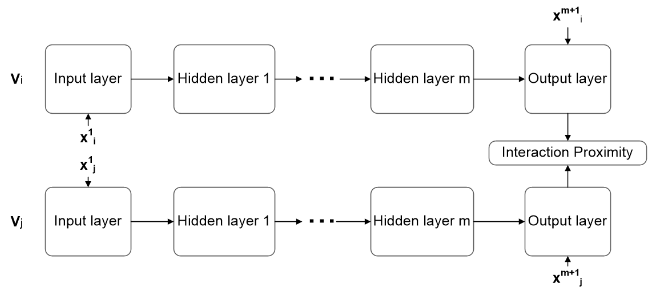

Fig. 3 GWNN model for each network snapshot embedding: The model input is X^1, and the output is X^{m+1}, where $X^1_{[:,i]}$ and $X^{m+1}_{[:,i]}$ are the i-th column of X^1 and X^{m+1} respectively (with the dimension $n \times 1$); we train the model and update the parameters through Formula 3.

4.2.2 LSTM layer

LSTMs [14] is an improved Recurrent Neural Network (RNNs). In order to solve the problem that RNNs cannot deal with long-distance dependence, LSTMs were proposed with forgetting units, which is designed to give the memory cells to determine when to forget information. The LSTM functions defined below:

$$\begin{cases} f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C^t = f_t * C_{t-1} + i^t * \tilde{C}_t \\ o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t = o_t * \tanh(C_t) \end{cases} \qquad (4)$$

For the LSTM, the computational process can be treated as a black box. The current x_t, previous hidden state h_{t-1} and cell state C_{t-1} are fed into LSTM, and they merge three inputs and compute the current hidden state h_t and new cell state C_t. This mechanism can effectively preserve historical information for each node.

In this paper, we adopt LSTM to predict new links. The input of the model is $\{Z_1, \ldots, Z_T\}$, where each Z_t with $t \in \{1, \ldots, T\}$ represents the output of the GWNN at the t time snapshot. We start with using $\{Z_1, \ldots, Z_{T-1}\}$ as the training sample and use Z_T for labelling. The model output is $\hat{Z}_T$ and an overview of our LSTM Training framework is described in Figure 4. We use square loss function as objective function to train model 5. After training, we shift the window one step towards the future to obtain the vector representation of each node for last snapshot. Then we train a downstream support vector machine classifier to evaluate link prediction.

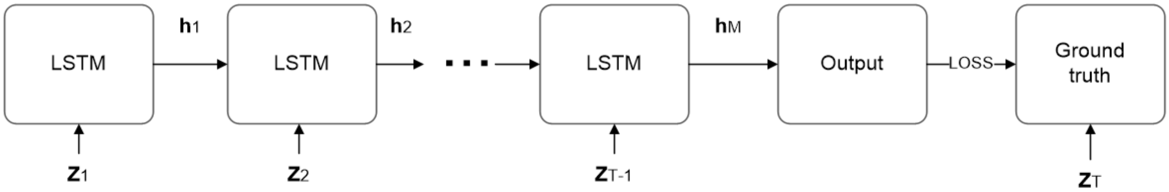

Fig. 4 Training framework for LSTM

$$Loss = (\hat{Z}_T - Z_T)^2 \tag{5}$$

5 Experiments

In this section, we introduce the datasets and baseline models, which use to validate the THS-GWNN's effectiveness in link prediction of temporal networks.

5.1 Datasets

We select four social temporal networks in the KONECT project [1)], which is two undirected temporal networks (Facebook friendships and Hep-Ph) and two directed networks (Digg and Facebook wall posts). Their statistic properties are shown in Table 1.

Table 1 The statistics for the datasets.

Network	#Nodes	#Links	Clustering coefficient	Format
Facebook friendships	63,731	817,035	14.8%	undirected
Hep-Ph	28,093	4,596,803	28.0%	undirected
Digg	30,398	87,627	0.56%	directed
Facebook wall posts	46,952	876,993	8.51%	directed

- The Facebook friendships dataset contains friendship data of Facebook users. The nodes represent users and edges are friendship between two users. The dataset is not complete and contains a very small subset of the total Facebook friendship network. We divide it by year and denote them as F_1 to F_5 for our experiments.
- The arXiv hep-ph dataset is the collaboration network of authors of scientific papers. The nodes represent authors and edges represent common publications. Timestamps represent the date of a publication. The dataset contains 10 years (1991 - 2002) of data. We select 5 years (1995 - 1999) and denote them as A_1 to A_5 and each snapshot contains a one-year network structure for our experiment.
- The Digg dataset is the reply network of the social news website Digg. The nodes represent users, and edges represent a user replied to another user. The dataset contains sixteen-days records, and we divide it by day. We evenly merged it into five snapshots by day and denote it as D_1 to D_5.
- The Facebook wall posts dataset is a small subset of posts to other user's wall on Facebook. The nodes represent Facebook users, and each edge is one post. The dataset contains 6 years (2004 - 2009) of data. For our experiment, we combined 2004 and 2005 data into one network snapshot and defined it as W_1. The rest of the data is defined as W_2 to W_5 by year and each snapshot contains a one-year network structure.

For our experiments on the above datasets, the last snapshot was used as ground-truth of network inference and the other snapshots was used to train the model.

5.2 Evaluation metric and baseline models

In our paper, we adopt the area under the receiver operating curve (AUC) [22] to evaluate the performance of different methods. The AUC relates to the sensitivity (true positive rate) and the specificity (true negative rate) of a classifier. This metric is strictly bounded between 0 and 1. The larger the AUC is, the better the model performs. We compare THS-GWNN with the following three baseline models.

- **STEP** [5]: STEP considers both spatial and temporal feathers at the same time, which utilises a joint matrix factorisation algorithm to simultaneously learn the spa-

1) http://konect.uni-koblenz.de/

tial and temporal constraints to model network evolution.

- **T-GCN** [30]: T-GCN combines with the graph convolutional network (GCN) to capture spatial dependence and the gated recurrent unit (GRU) to capture temporal dependence.
- **NetWalk** [23]: NetWalk model updates the network representation dynamically as the network evolves by clique embedding, it focuses on anomaly detection and we adopt its representation vector for link prediction.

Parameter settings. In the last snapshot, the connected links are often very sparse. Hence, we randomly generate the number of non-linked edges smaller than twice linked edges to ensure data balance in the evaluation process [7]. For the dimension of the embedded vector, we set the Hep-Ph (28,093 nodes) and the Digg (30,398 nodes) are 256 dimensions and the Facebook wall posts dataset (46,952 nodes) and Facebook friendships (63,731 nodes) are 512 dimensions. If we increase or decrease the dimensions, the performance remains the same or even becomes worse. For different datasets, the parameters for baselines are tuned to be optimal. The BCGD method is only for undirected networks. For the directed networks, we transfer the adjacency matrix of directed networks to undirected networks by $(A^T + A)/2$ [5]. Other settings include: the learning rate of the model is set as 0.0001; the number of hops K is set as 3; the temporal window L is set as 3; the m for layers of the GWNN is set as 5; the number of neighbors sampled in layer 1: A_t^1 is set as 500. the decaying exponential γ is set as 0.8. For the result of our experiment, we carried out five times independently and reported the average AUC values for each dataset.

5.3 Experimental results

For experiments, we compare the performance of three baselines on four temporal networks for link prediction. We first embed each vertex into a vector at each snapshot. For each dataset, we divide by timestamp, and the last snapshot was used as ground-truth of network inference, and the previous snapshots are used to train the THS-GWNN model. After training, we shift the window one step towards the future to obtain the vector representation of each node for last snapshot. Last, we use the obtained representations to predict the network structure.

Table 2 Prediction results for the four datasets (AUC value).

Model	Hep-Ph	Digg	Facebook wall posts	Facebook friendships
STEP	0.61	0.74	0.76	0.57
NetWalk	0.69	0.71	0.74	0.70
T-GCN	0.70	0.75	0.72	0.71
THS-GWNN	**0.74**	**0.82**	**0.78**	**0.73**

Table 2 compares AUCs over the four datasets. Compared with the baselines, our method, THS-GWNN, achieves the best performance. THS-GWNN adopts the THS algorithm to sampling $\Gamma(v, K, t_{Sta}, t_{End})$ for each node v, which can better capture both spatial and temporal features for each node. It incorporates a decaying exponential γ to assign the more sampled nodes to the fewer hops and the closer snapshots, which can better preserve the evolving behavior of temporal networks. THS-GWNN also adopts GWNN to embed node spatial-temporal features, this can better capture the nonlinear network attributes due to it is a deep model [6]. As such, it has advantages over the above baseline models.

5.4 Ablation study

We conducted an ablation study on the arXiv hep-ph dataset and the Digg dataset in this section. (1) We replaced the GWNN unit with the GCN [15] (THS-GCN) unit to verify the performance of the THS-GWNN model. The experimental results show the average AUC value of THS-GWNN is 3% higher than THS-GCN on the Digg dataset, 2% higher than THS-GCN on the arXiv hep-ph dataset. (2) We replaced the THS sampling algorithm with the PinSage [28] sampling strategy (PinSage-GWNN) to verify the performance of the THS-GWNN model. The sampling strategy uses short random walks to sample k-hop neighbor nodes as features. The experimental results show the average AUC value of THS-GWNN is 5% higher than PinSage-GWNN on the Digg dataset, 4% higher than PinSage-GWNN on the arXiv hep-ph dataset. The reason may be that the PinSage sampling strategy only extracts the spatial features of the network rather than the temporal features of the network.

5.5 Parameter sensitivity analysis

We conduct parameter sensitivity analysis in this section, and the results are shown in Figure 5. Specifically, we estimate

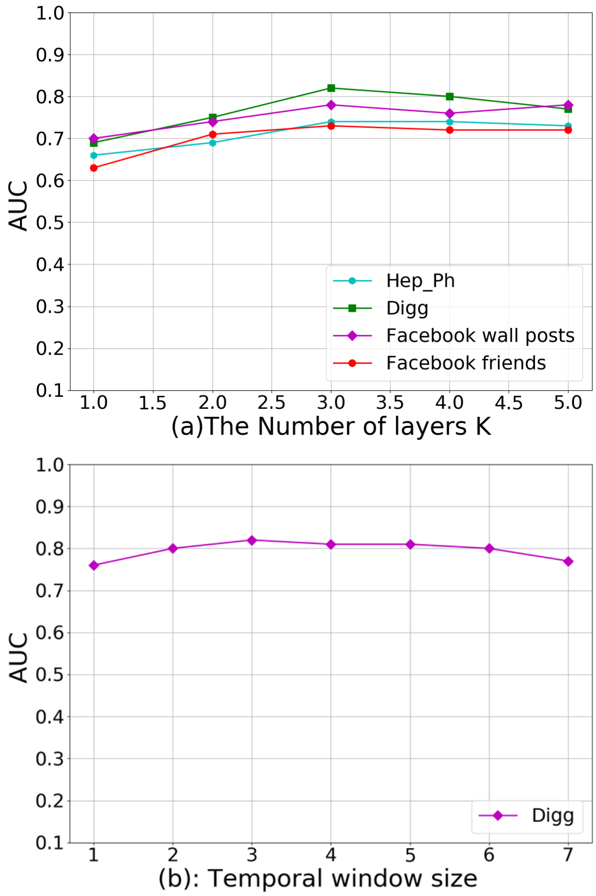

Fig. 5 Experiments on parameters sensitivity. (a) THS-GWNN's performance on four datasets when increasing K. (b) THS-GWNN's performance on the Digg dataset when increasing L.

how different the number of layers K and the temporal window L can affect the link prediction results.

- The number of layers K. We vary the layer of each snapshot from 1 to 5 to verify the validity of this parameter. When this parameter is verified, other parameters are set with their default values. As can be seen from the Figure 5a, the performance continues to increase as K increases from 1 to 3. The reason might be that the more K-hop neighbours the current node has, the more it can represent the current node. The best result is obtained at $K = 3$, after which the performance decreases slightly or remains unchanged as K continues to increase. The reason might be that the further away from the current node, the less information there is about the current node.
- Temporal window size L. Since the Digg dataset contains sixteen-days records, it will generates 16 snapshots if we split it by day. It has more snapshots than other datasets, so we select the Digg dataset to conduct sensitivity analysis for the parameter L. We vary the window size from 1 to 7 to verify the validity of this parameter. When this parameter is verified, other parameters are set with their default values. As can be seen from the Figure 5b, the performance continues to increase as L increases from 1 to 3. The reason might be that the closer snapshot is to the current snapshot, the more information about the current snapshot. The best results is obtained when $L = 3$, after which the accuracy no longer increases, when L continuously increases.

6 Conclusions

In this paper, we have proposed an effective framework, THS-GWNN, for link prediction in temporal networks, which captures both spatial and temporal evolution features of the networks. In particular, we proposed the THS algorithm to extract both spatial and temporal features in each snapshot to model network evolution. It incorporates a decaying exponential γ to assign the more sampled nodes to the fewer hops and the closer snaphsots, which can better preserve the evolving behavior of temporal networks. We then adopt graph wavelet neural networks to embed the spatial-temporal features into vectors, which can better capture the nonlinear network attributes. During the link prediction phase, we use long short term memory networks to capture the time dependence among network snapshots. Experiments demonstrate the effectiveness of our THS-GWNN model and it achieves significant gains than the baselines models.

Our future work will study merging centrality (e.g., degree centrality) and topology information to improve the feature sampling strategy for temporal networks and study the transferability of our model on various tasks. In addition, we also study how to aggregate other types of features [33] and pay more attention to time and space complexity by conducting more comprehensive experiments.

References

1. Xuan Q, Zhang Z Y, Fu C, et al. Social synchrony on complex networks. IEEE Transactions on Cybernetics, 2017, 48(5): 1420-1431.

2. Pavlopoulos G A, Wegener A L, Schneider R. A survey of visualization tools for biological network analysis. Biodata Mining, 2008, 1(1): 12.
3. Lv Y, Duan Y, Kang W, et al. Traffic flow prediction with big data: a deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 2014, 16(2): 865-873.
4. Martínez V, Berzal F, Cubero J C. A survey of link prediction in complex networks. ACM Computing Surveys, 2016, 49(4): 1-33.
5. Chen H, Li J. Exploiting structural and temporal evolution in dynamic link prediction. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018, 427-436.
6. Li T, Zhang J, Philip S Y, et al. Deep dynamic network embedding for link prediction. IEEE Access, 2018, 6: 29219-29230.
7. Zhu L, Guo D, Yin J, et al. Scalable temporal latent space inference for link prediction in dynamic social networks. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(10): 2765-2777.
8. Goyal, Palash, Chhetri, Sujit Rokka, Canedo, Arquimedes. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. Knowledge-Based Systems, 2020, 187:104816.
9. Dunlavy D M, Kolda T G, Acar E. Temporal link prediction using matrix and tensor factorizations. ACM Transactions on Knowledge Discovery from Data, 2011, 5(2): 1-27.
10. Yu W, Cheng W, Aggarwal C C, et al. Link Prediction with Spatial and Temporal Consistency in Dynamic Networks. In: Proceedings of the International Joint Conference on Artificial Intelligence, 2017: 3343-3349.
11. Güneş İ, Gündüz-Öğüdücü Ş, Çataltepe Z. Link prediction using time series of neighborhood-based node similarity scores. Data Mining and Knowledge Discovery, 2016, 30(1): 147-180.
12. Yu L, Liu H. Feature selection for high-dimensional data: A fast correlation-based filter solution.In: Proceedings of the 20th International Conference on Machine Learning, 2003: 856-863.
13. Xu, B, Shen, H, Cao, Q, Qiu, Y, Cheng, X. Graph wavelet neural network. In: Proceedings of the International Conference on Learning Representations, 2019: 1-13.
14. Hochreiter, S, Schmidhuber, J. Long short-term memory. Neural Computation, 1987, 9(8): 1735-1780.
15. Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.
16. Perozzi, B, Al-Rfou, R, Skiena, S. Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014: 701-710.
17. Grover, A, Leskovec, J. node2vec: Scalable feature learning for networks, In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016: 855-864.
18. Wang, D, Cui, P, Zhu, W. Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016: 1225-1234.
19. Cho, K, Van Merriënboer, B, Bahdanau, D, Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014.
20. R N Lichtenwalter, J T Lussier, N V Chawla. New perspectives and methods in link prediction. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010: 243-252.
21. Sankar, Aravind, Wu, Yanhong, Gou, Liang, Zhang, Wei, Yang, Hao. DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks. Proceedings of the 13th International Conference on Web Search and Data Mining, 2020:519–527.
22. Huang J, Ling C X. Using AUC and accuracy in evaluating learning algorithms. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(3): 299-310.
23. Yu, Wenchao, Cheng, Wei, Aggarwal, Charu C, Zhang, Kai, Chen, Haifeng, Wang, Wei. NetWalk: A Flexible Deep Embedding Approach for Anomaly Detection in Dynamic Networks. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018:2672–2681.
24. Chen J, Zhang J, Xu X, et al. E-lstm-d: A deep learning framework for dynamic network link prediction. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019.
25. Zhou T, Lü L, Zhang Y C. Predicting missing links via local information. The European Physical Journal B, 2009, 71(4): 623-630.
26. Munasinghe L. Time-aware methods for link prediction in social networks. Ph. D. thesis, The Graduate University for Advanced Studies, 2013.
27. Newman M E J. Clustering and preferential attachment in growing networks. Physical Review E, 2001, 64(2): 025102.
28. Ying, Rex, et al. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In: Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2018: 974-983.
29. Yu, Bing, Yin, Haoteng, Zhu, Zhanxing. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018:3634–3640.
30. L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. IEEE Transactions on Intelligent Transportation Systems, 2019, 1-11.
31. Singer, Uriel, Guy, Ido, Radinsky, Kira. Node Embedding over Temporal Graphs. Proceedings of the 2019 International Joint Conference on Artificial Intelligence, 2019:4605-4612.
32. Li, Yaguang, Yu, Rose, Shahabi, Cyrus, Liu, Yan. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. Proceedings of the 2018 International Conference on Learning Representations, 2018:1-16.

33. Pang J, Zhang Y. DeepCity: A feature learning framework for mining location check-ins. Proceedings of the 11th International AAAI Conference on Web and Social Media, 2017: 652-655.

Xian Mo is currently a PhD student at School of Computer and Information Science, Southwest University, China. He received her master degree from Chongqing University of Technology in 2016. His principal research interest is in representation learning for temporal networks. During the period of learning, he presided two research programs of the Chongqing graduate Research and Innovation Project and the Fundamental Research Funds for the Central Universities.

Jun Pang is currently a Senior Researcher at Faculty of Science, Technology and Medicine & Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg. He received his PhD degree from Vrije Universiteit Amsterdam in 2004. His principal research interest is in formal methods, social media mining, security and privacy. He is a recipient of CMSB Best Paper Award, 2019, APBC, Best Poster Award, 2018, etc. He has published more than 180 journal and conference papers.

Zhiming Liu is currently a Professor at School of Computer and Information Science, Southwest University, China (the Recruitment Program of Global Experts). He received his PhD degree from University of Warwick in 1991. His principal research interest focuses on formal methods. He received a second prize of Natural Science Category of the Macau SAR Natural Science Award in 2012. He has published more than 150 journal and conference papers.