Journal of Universal Computer Science, vol. 29, no. 5 (2023), 470-490
submitted: 22/12/2022, accepted: 28/3/2023, appeared: 28/5/2023 CC BY-ND 4.0

Topological Similarity and Centrality Driven Hybrid Deep Learning for Temporal Link Prediction

Abubakhari Sserwadda
(Computer Engineering Department, Istanbul Technical University, Istanbul, Turkey
https://orcid.org/0000-0001-9150-5298, sserwadda@itu.edu.tr)

Alper Ozcan
(Computer Engineering Department, Akdeniz University, Antalya, Turkey
https://orcid.org/0000-0002-5999-1203, alperozcan@akdeniz.edu.tr)

Yusuf Yaslan
(Computer Engineering Department, Istanbul Technical University, Istanbul, Turkey
https://orcid.org/0000-0001-8038-948X, yyaslan@itu.edu.tr)

Abstract: Several real-world phenomena, including social, communication, transportation, and biological networks, can be efficiently expressed as graphs. This enables the deployment of graph algorithms to infer information from such complex network interactions to enhance graph applications' accuracy, including link prediction, node classification, and clustering. However, the large size and complexity of the network data limit the efficiency of the learning algorithms in making decisions from such graph datasets. To overcome these limitations, graph embedding techniques are usually adopted. However, many studies not only assume static networks but also pay less attention to preserving the network topological and centrality information, which information is key in analyzing networks. In order to fill these gaps, we propose a novel end-to-end unified **T**opological **S**imilarity and **C**entrality driven Hybrid Deep Learning model for **T**emporal **L**ink **P**rediction (TSC-TLP). First, we extract topological similarity and centrality-based features from the raw networks. Next, we systematically aggregate these topological and centrality features to act as inputs for the encoder. In addition, we leverage the long short-term memory (LSTM) layer to learn the underlying temporal information in the graph snapshots. Lastly, we impose topological similarity and centrality constraints on the model learning to enforce preserving of topological structure and node centrality role of the input graphs in the learned embeddings. The proposed TSC-TLP is tested on 3 real-world temporal social networks. On average, it exhibits a 4% improvement in link prediction accuracy and a 37% reduction in MSE on centrality prediction over the best benchmark.

Keywords: Topological similarity, centrality, embedding learning, link prediction, deep geometric learning
Categories: G.2.2, I.2.6, I.5.1, I.5.4
DOI: 10.3897/jucs.99169

1 Introduction

Numerous real-world scenario data, including social, transport, and biological networks, can be conveniently modeled as graphs. This enables the application of graph algorithms and tools to effectively analyze large and complex network interactions so as to meet the needs of several graph applications, including node classification [Bhagat et al., 2011], community detection [Newman and Girvan, 2004], node clustering [Nie et al., 2017], and node recommendation [Wang et al., 2010]. A number of real-world problems, including disease outbreak prediction [Dallas et al., 2019], detection of spam emails [Huang and Zeng, 2006], prediction of user interaction in social networks [Mallek, 2018] and protein-to-protein interaction in biological networks [Koutrouli et al., 2020] can be modeled as link prediction problems. In order to analyze and perform such relevant predictions on networks, it is necessary to understand how node connectivity varies across networks. Among the vital measures for computing and interpreting these node connectives includes centrality [Freeman et al., 1991, Klein, 2010], and topological similarity [Newman, 2004, Salton and McGill, 1983]. Node centrality serves as a powerful tool in several scenarios of network analysis. For example, in [Nobre et al., 2022], the WhatsApp user centrality measure was used to analyze the spread of misinformation on WhatsApp, where it was realized that WhatsApp users with higher centrality often contribute most to coverage of communities in terms of content diversity. Used for exploring brain regions of interest and connections that are so significant for communication flow in the brain [Kwon et al., 2019], identification of key nodes in road transport networks to boost the safety of traffic operations [Liu et al., 2019], predicting the spread of epidemics and diseases including AIDS [Bucur and Holme, 2020], preserving of most influential brain regions during brain data generation [Sserwadda and Rekik, 2021] and changes in node centralities can be used to monitor changes in dynamic networks and predict abnormal and suspicious events in a network [Aleskerov and Shvydun, 2018]. On the other hand, several research has highlighted the role of topological similarity-based features for different tasks, including link prediction [Li et al., 2018, Coşkun and Koyutürk, 2021] and recommendation [Li et al., 2014].

However, the large size and complexity of the network interaction datasets limit the efficiency of learning algorithms when deployed on such datasets. To circumvent this, embedding learning, a technique that projects a network into a low dimensional space, is widely adopted [Yan et al., 2006]. Traditionally, researchers deploy random walk [Sajjad et al., 2020], matrix factorization [Zhu et al., 2016a], and shallow dimension reduction techniques, including LDA [Fisher, 1936], and PCA [Yan et al., 2006]. However, these methods rely on engineered node features, and thus they are less efficient at preserving the graph structure of large and complex networks.

On the other hand, given the promising results of deep learning operations on images in recent works, researchers have extended deep learning approaches to graphs and manifolds, a technique referred to as geometric deep learning [Monti et al., 2017]. However, unlike in images, where data is defined on the Euclidean domain, graphs and manifolds are non-Euclidean; that is, the shortest distance between any two nodes is not necessarily a straight line. Moreover, such non-Euclidean data might have self-connections, infinite curvature, and different dimensions depending on the view scale, direction, and location. For graph data, in particular, nodes often have varying sizes of local neighboring nodes, with each node having links with other distant individual nodes. This renders deployment of the typical spatially localized convolution operation on such dataset non-obvious as Euclidean geometric-based properties and space representation may not apply on graphs.

Despite the above challenges, several researchers have explored means for extending deep learning approaches to graphs and manifolds. In [Wu et al., 2020, Zhou et al., 2020], reviews of these approaches are presented. Research in [Kipf and Welling, 2017] presents GCNs, the pioneering work for extending the image-based convolution operation on graphs. Later several variants of GCNs were proposed to tackle the graph embedding problem [Pareja et al., 2020, Velickovic et al., 2018]). However, the typical GCN convolution operation can only preserve the local connectivity information of the network and not the global structure of the graph. Although multilayer GCNS can be leveraged to capture high-order information. The increasing number of GCN layers exposes the learning model to overfitting. Thus, a maximum of only 3 GCN layers is usually adopted. More to this, although several real-world scenarios have evolving graph data, the majority of research on learning graph embedding, including [Ma et al., 2020, Qiao and Hu, 2020], assumes static graphs neglecting the network temporal evolution information. Most importantly, researchers pay less attention to preserving the underlying vital node centrality roles and topological structure of networks while learning network endings, yet they are key properties in characterizing networks. To fill the above gaps, we propose an end-to-end Topological Similarity and Centrality driven hybrid deep learning model for Temporal Link Prediction (TSC-TLP), that efficiently utilizes topological similarity and centrality-based features as inputs, coupled with the LSTM (long short-term memory) layer to capture the underlying temporal evolution information in the graph snapshots. In addition, we impose topological similarity and centrality constraints on the model learning to enforce preserving the topological structure of the input graphs in the learned embeddings.

The major contributions of our proposed TSC-TLP can be summarized as follows:

1. Extracting and systematically aggregating topological and centrality-based features as model inputs.

2. Imposing topological and centrality constraints on the model learning to enforce preserving of the topological network structure and node centrality roles in learned embeddings.

3. Integrating all the model modules in an end-to-end unified framework to minimize the accumulation of errors common with subdivided tasks.

The remaining part of the paper is organized as follows. Section 2 reviews the literature, preliminaries, and problem definition presented in Section 3. Section 4 details the methods used in this study, experiments, and test results are presented in Section 5. Section 6 analyzes the experimental results. Finally, the paper is concluded in Section 7.

2 Related Work

In this section, the related methods and studies presented in the literature are discussed. The literature review is classified into Matrix factorization-based, Random walk-based, and Deep learning-based graph embedding methods.

2.1 Matrix factorization-based network embedding learning

In matrix factorization-based techniques, the graph is represented as a matrix (e.g., adjacency matrix), and the matrix is then decomposed to yield a low-dimension node representation. For instance, in [Zhu et al., 2016b], a temporal graph latent space is learned via a negative matrix factorization approach by adopting local and incremental block-coordinate gradient descent algorithms. While [Zhu et al., 2016a] adopts a traditional matrix factorization to learn network embeddings and Markov processes to model the temporal network. Most recently, [Aghdam and Zanjani, 2021] proposed a novel regularized asymmetric non-negative matrix factorization (RANMF) that extracts pairwise similarity between feature vectors while learning embeddings to better the performance of a wide range of applications. They imposed similarity constraints on the cost function as regularization variables to fasten convergence. However, Matrix factorization approaches are characterized by large space and computational requirements due to storing large matrices, especially for large networks. Moreover, matrix factorization assumes that the target matrix is linearly decomposable, which may not always hold.

2.2 Random walk-based network embedding learning

To overcome the above challenges associated with matrix factorization, several works on learning graph embeddings deploy random walks to capture node neighborhood connectivity information. In random walk-based network embedding techniques, basically, a graph is decomposed into random paths and modeled as the frequency of random walks in it. For a graph G, taking the i^{th} path as a triplet $< p_i^s, p_i^e, l_i >$, where p_i^s and p_i^e denote the starting and ending vertices of the paths respectively and l_i for the length of the path. Graph G is then modeled as a d dimensional vector with the i^{th} element denoting the frequency of G's i^{th} triplet. For example, [Chen et al., 2018] proposes to compress the input graph before the embedding stage and adopt random walks to capture higher-order graph connection information. DeepWalk [Perozzi et al., 2014] utilizes random walks and Skip-Gram to obtain local information for learning the node embeddings by treating the distribution of nodes in the short random walks to be almost similar to that of words in natural language. Nonetheless, random walks basically extract only the local proximity information within a path but not the global network information, and there is no strategy to guarantee optimal node or edge sampling.

2.3 Deep learning-based graph embedding methods

The introduction of graph convolution networks (GCN) enhanced research in deep geometric learning. Recently, quite a large number of GCN variants have been proposed. Intuitively, the graph convolution operation computes the node's new features as the weighted average of its neighbors and itself. Like the typical convolution operation, a single convolution layer preserves the immediate connectivity information of the node. In practice, multilayer GCNs are deployed to capture high-order connectivity information in a network. However, increasing the number of layers exposes the typical GCN to overfitting [NarasingaRao et al., 2018]. Thus, the GCN is limited to only capturing the local node connectivity information but not the global network information. Moreover, the GCN was proposed to handle static graphs, yet in the real world, many networks are evolving, with new interactions between nodes over time. On the other hand, [Pareja et al., 2020] proposed EvolveGCN, a dynamic variant of GCN that adapts the GCN model along the temporal dimension without resorting to node embeddings. It deploys RNN to evolve the GCN parameters, thereby capturing the dynamism of network sequences. In addition, [Chen et al., 2021] deploys GCN to extract the network structural features and LSTM to extract the temporal network features. Another variant of GCN that implicitly specifies different weights for the neighboring nodes, based on their importance on the reference node, is proposed in [Velickovic et al., 2018]. It stacks layers enabling nodes to attend over features of their neighborhoods. Furthermore, motivated by the efficiency of the variational autoencoder in generating data by exploring variations in the input data and the success of the LSTM in learning long-term dependencies in sequential data, [Chen et al., 2019] proposed an end-to-end encoder-LSTM-decoder architecture for link prediction in dynamic graphs, the first of its kind to combine the LSTM and encoder-decoder architecture for the link prediction task. The model exhibited great efficiency at predicting not only links that will appear but also those that will disappear. On the other hand, research in [Shaw and Jebara, 2009, Sserwadda and Rekik, 2021, Bessadok et al., 2020] demonstrates the significance of introducing graph structure and property preserving constraints into low dimensional learning algorithms, where such learning constraints are observed to boost the quality of learned embeddings. Motivated by the above findings, we propose an end-to-end TSC-TLP that leverages encoder-decoder architecture to efficiently learn the underlying variations in the input data, with an LSTM layer to capture the temporal information. In addition, we impose the topological similarity and centrality constraints on the model learning to further ensure that the topological and structural centrality of input graphs are preserved in the learned embeddings so as to have a more accurate prediction of the target network.

3 Problem Definition and Preliminaries

In this section, we detail the steps for the proposed TSC-TLP.

3.1 Problem Definition

In this paper, matrices are denoted by boldface capital letters, e.g., **C**, vectors are marked by boldface lowercase letters, e.g., **c**, and scalars are denoted by lowercase letters, e.g., c. The major mathematical notations used in this work are summarized in Table 1 for easy reference. For an input graph $\mathbf{G} = (V, E, W)$, where $V = v_1, v_2, ...v_n$ is the set of n nodes, $E = \{e_{i,j}\}_{i,j=1}^{n}$ denotes edges, where $e_{i,j} \in E$ links node $v_i \in V$ to $v_j \in V$

Notation	Definition
N	total number of subjects in the population
n_r	total number of nodes in a network
$\mathbf{v}_i$	input node vector for the i^{th} node $\in \mathbb{R}^n$
$\widetilde{\mathbf{v}}_i$	embedding vector for i^{th} input node vector $\mathbf{v}_i$, $\in \mathbb{R}^d$, where $d < n$
$\widehat{\mathbf{v}}_i$	predicted vector for i^{th} input node vector $\mathbf{v}_i$, $\in \mathbb{R}^n$
$\mathbf{A}_{ij}$	adjacency matrix for node pair i and j $\in \mathbb{R}^{n \times n}$
$\widehat{\mathbf{A}}_{ij}$	predicted adjacency matrix for node pair i and j $\in \mathbb{R}^{n \times n}$
m	topological similarity value
$\mathbf{M}$	topological similarity matrix
s	node strength value
$\mathbf{S}$	strength matrix
$\mathbf{D}$	degree matrix
λ	constant value
γ	constant value

Table 1: Key mathematical notations used in this paper.

and an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. $\mathbf{A}_{i,j} = 0$ if nodes V_i and V_j are not linked by any edge and $\mathbf{A}_{i,j} > 0$ otherwise. We aim at effectively learning network embeddings while preserving the topological similarity and centrality network information so as to boost the accuracy of graph applications, including link prediction that relies on the quality of learned embeddings. For a graph $\mathbf{G} = (V, E)$, network embedding aims at learning a mapping function f:$\mathbf{v_i} \longrightarrow \widetilde{\mathbf{v}}_i \in \mathbb{R}^d$, where $d < \|\mathbf{v}\|$. Precisely for dynamic graphs, taking $\mathcal{G}$= $\{\mathbf{G}_1, \mathbf{G}_2, ...\mathbf{G}_T\}$ as an evolution of graph of the **G**, where $\mathbf{G}_t$, represents the graph shot at any time t. We aim at learning a function f_t for mapping each node vector **v** in series of low dimensional vector space $\{\widetilde{\mathbf{v}}_1, ...\widetilde{\mathbf{v}}_t\}$, where $\widetilde{\mathbf{v}}_t$ is the embedding vector of node **v** at time t. $\widetilde{\mathbf{v}}_t = f_t(\mathbf{v}_1, ., ., \mathbf{v}_t)$. (i.e., the embedding function f_t at each time step exploits graph information from previous time steps to efficiently capture the network dynamics during embedding learning.)

3.2 Topological similarity-based features

We explored four popular topological similarity measures, including common neighbor [Newman, 2001], Jaccard Index [Jaccard, 1912], Adamic-Adar [Adamic and Adar, 2003], and Salton Index [Salton and McGill, 1983]. For a node i, let $\Gamma(i) \subseteq V$ be a set of neighbours of i, $(i, j) \in E$. m and **M** are topological similarity values and matrix respectively.

3.2.1 Common Neighbor

Common neighbor (CN) considers two nodes to be more related if connected to the same set of other vertices. The number of common neighbors of nodes $i \in V$ and $j \in V$ is defined as:

$$m_{CN}(ij) = |\Gamma(i) \cap \Gamma(j)|. \tag{1}$$

Its matrix form can be computed as;

$$M_{CN} = (\mathbf{A}^2). \tag{2}$$

3.2.2 Adamic-Adar

The Adamic-Ader (AA) topological similarity measure counts common neighbors by assigning more weights to less-connected common neighbors. (i.e., a common neighbor that is unique for a few nodes only is considered to be more important than a hub).

$$m_{AA} = \sum_{z \in \Gamma(i) \cap \Gamma(j)} \frac{1}{\log k_z}. \tag{3}$$

In a matrix form, it can be computed as;

$$M_{AA} = \mathbf{A} log(\mathbf{D}^{-1})\mathbf{A} \tag{4}$$

3.2.3 Salton Index (cosine similarity)

The Salton Index (SI) measures the overlap size by degrees of any two nodes. (i.e., the cosine angle between vectors of nodes in the adjacency matrix)

$$m_{SI}(ij) = \frac{|\Gamma(i) \cap \Gamma(j)|}{\sqrt{|\Gamma(i) \times \Gamma(j)|}}. \tag{5}$$

Its matrix form can be formulated as;

$$M_{SI} = \mathbf{A}^2 \oslash \mathbf{D}. \tag{6}$$

3.2.4 Jaccard Index

The Jaccard Index (JI) computes the proportion of common neighbors in the total number of neighbors.

$$m_{JI}(ij) = \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|}. \tag{7}$$

In a matrix form, it can be formulated as;

$$M_{SI} = \mathbf{A}^2 \oslash (\mathbf{AN} + \mathbf{NA} - \mathbf{A^2}). \tag{8}$$

Here, N denotes an all-ones matrix with the same dimension as **A**, and $\oslash$ denotes element-wise (Hadamard) division.

3.3 Topological strength centrality

Node degree quantifies the number of links incident to a node in a binary network, thus the degree k_i of node i in a network with other nodes $j = 1...N - 1$ can be computed as;

$$k_i = \sum_{j \neq i} \mathbf{A}_{ij}. \tag{9}$$

Node strength is the analog of node degree in a weighted network. It quantifies the connectivity weights of the edges incident to each node in an undirected network. The strength s_i of the node i can be computed as;

$$s_i = \sum_{j \neq i} \mathbf{W}_{ij}. \tag{10}$$

where $\mathbf{W}_{ij}$ is the weight of the edge connecting nodes i and j. By summing all edge weights attached to the reference node, the node strength indexes the node's global connectivity [Geethanjali, 2015]. We use **S** to denote the topological strength matrix.

4 Proposed TSC-TLP model

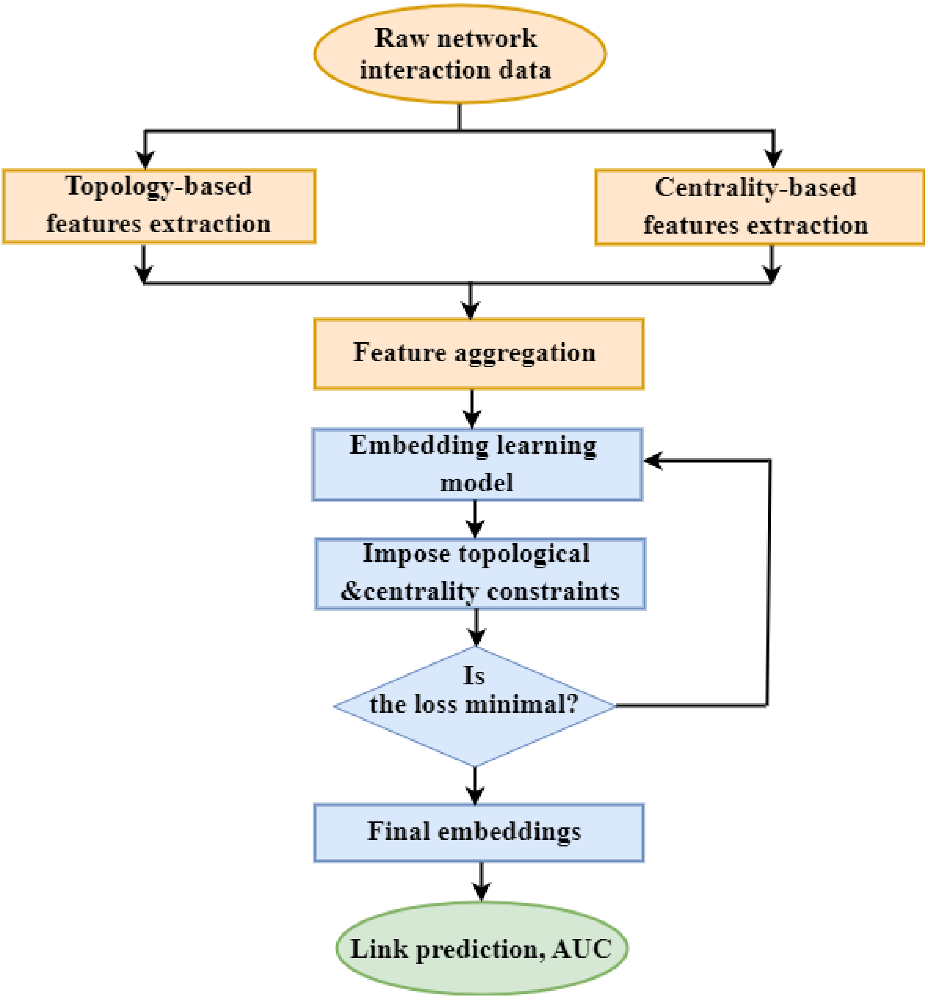

Raw network interaction data
Topology-based features extraction
Centrality-based features extraction
Feature aggregation
Embedding learning model
Impose topological &centrality constraints
Is the loss minimal?
Final embeddings
Link prediction, AUC

Figure 1: The flow chart of the proposed TSC-TLP framework. Topology and centrality-based features are extracted from the raw social network interaction datasets as guided by Eqs.1-7 and Eq.10, respectively. These features are then aggregated and fed to the deep learning model to learn the network embeddings. The topological and centrality constraints are imposed on the model learning loss 15 to enforce the preservation of network topology and structural central of nodes. The model is retrained until the training loss reduces to a certain minimum level. Then final embeddings for time t are obtained and used to predict the network at time $t + 1$. The AUC score (Eqs.16-17) is computed and reported as the link prediction accuracy

Figure 1 summarizes the major steps of the proposed TSC-TLP framework. The input to the model is the aggregation of the topology similarity and strength centrality matrices of previous graphs by a summation operation, and its output is the graph at the next time step. The model captures the complex interactions among graph nodes at each time step and throughout the time steps while preserving the topological similarity and structural centrality of the network during embedding learning. The aggregated topological similarity and strength centrality features are fed to the fully connected auto-encoder that outputs a low dimensional representation of a node u, as shown in the equation below.

$$y_{u_t}^{h} = f_a(W^{h} y_{u_t}^{h-1} + b^{h}). \tag{11}$$

where h is the output layer for the fully connected auto-encoder. f_a is the activation function, W is the learnable weight matrix, and b denotes the bias. Then, the learned low representation is passed through the LSTM network as in the following equation.

$$y_{u_t}^{h+1} = O_{u_t}^{h+1} * tanh(C_{u_t}^{h+1}) \tag{12}$$

$$O_{u_t}^{h+1} = \sigma_{u_t}^{h+1}(W^{j+1}[y_{u_t-}^{h+1}, y_{u_t}^{h}] + b_o^{h+1}). \tag{13}$$

The output of the LSTM network is then fed to a fully connected decoder. For the loss function, we follow the approach proposed in [Goyal et al., 2020], in which the model learns the embedding at time step t by optimizing the loss function shown in Eq.14 below.

$$\begin{gathered} \mathcal{L}_{t+n} = ||(\widehat{\mathbf{A}}_{t+n+1} - \mathbf{A}_{t+n+1}) \odot \boldsymbol{\mathcal{B}}||^2 \\ \text{where,} \quad \widehat{\mathbf{A}}_{t+n+1} = f(\mathbf{A}_t, ..., \mathbf{A}_{t+1}) \end{gathered} \tag{14}$$

$\boldsymbol{\mathcal{B}}$ is a weighting matrix to weight the reconstruction of observed edges higher than unobserved links as it is used in [Wang et al., 2016]. $\odot$ denotes element-wise product. Inspired by work in [Shaw and Jebara, 2009] who recommends that introducing network structure preserving constraints into dimensional reduction learning models yields more accurate representations of high dimensional data, as well as in [Sserwadda and Rekik, 2021, Bessadok et al., 2020] where related topological constraints enhanced the performance of the learning models. We propose to impose the topological similarity and strength constraints on the model learning so as to enforce preserving the topological and centrality structures of the input graphs in the learned embeddings. Thus, the total loss becomes;

$$\begin{aligned} \mathcal{L}_{t+n} =& ||(\widehat{\mathbf{A}}_{t+n+1} - \mathbf{A}_{t+n+1}) \odot \boldsymbol{\mathcal{B}}||^2 \\ &+ \lambda||(mm(\widehat{\mathbf{A}}_{t+n+1}) - mm(\mathbf{A}_{t+n+1}))||^2 \\ &+ \gamma||(ss(\widehat{\mathbf{A}}_{t+n+1}) - ss(\mathbf{A}_{t+n+1}))||^2 \end{aligned} \tag{15}$$

where $\boldsymbol{\mathcal{B}}$, a weighting matrix to weight the reconstruction of observed edges higher than unobserved links as it used in [Wang et al., 2016, Goyal et al., 2020], mm and ss are the topological similarity and strength centrality computation functions respectively, λ and γ regulate the weight attached to the topological similarity and strength centrality constraints respectively, and n as the window size parameter that regulates the length of the temporal patterns learned. Pseudocode 1 summarizes the flow of the procedures in the proposed TSC-TLP.

Algorithm 1: Pseudoce for the proposed TSC-TLP

```
Input: G = $(V, E)$
1 Compute Topological similarity and Centrality-based features using Eqs. 1-7
    and Eq. 10 respectively ;
2 Aggregate features;
3 Feed the aggregated features to the deep embedding learning model, and train
    the model following Eqs.11-14 ;
4 Predict the likelihood of the existence of a link between node pairs at time $t + 1$;
Output: Obtain AUC guided by Eqs.16-17.
```

We explored features for four popular topological similarity measures, including common neighbor [Newman, 2001], Jaccard Index [Jaccard, 1912], Adamic-Adar [Adamic and Adar, 2003], and Salton Index[Salton and McGill, 1983]. **Fig.** 2 illustrates the architecture of the proposed TSC-TLP model.

5 Experiments

To measure the quality of link prediction produced by TSC-TLP, we conducted experiments on 3 (three) real-world datasets, publicly available.

5.1 Datasets

The datasets used in this experiment include a student message network at the University of California (Irvine) [1], a communication network from Autonomous Systems (AS) [2], and a user interaction network from the stack exchange website Math Overflow (Maths) [3]. These datasets are detailed in Table 2.

5.2 Parameter settings

The embedding size is searched from {64,128,256} and 128 is selected, as it yielded the best results, number of epochs heuristically set to 50, Adam optimizer [Kingma and Ba, 2015] is used for the optimization of the learning models. The learning rate is searched from the range $\{0.001, 0.002, 0.003, 0.004, 0.005\}$ and 0.002 is selected as it offers the best results. A full batch is used in each training epoch. The hyper-parameters λ and γ that regulate the weights attached to the topological similarity and strength centrality constraints respectively are selected from the search range $\{0.1, 0.2..., 1\}$. The PyTorch deep graph library is used for the implementation of the models. All experiments are performed on Ubuntu 20.04.3 LTS, Nvidia GeForce GTX 1080Ti, 64GB RAM, 32 cores, and clock 33MHz GPU.

[1] http://konect.cc/networks/opsahl-ucsocial/

[2] http://snap.stanford.edu/data/as-733.html

[3] http://snap.stanford.edu/data/sx-mathoverflow.html

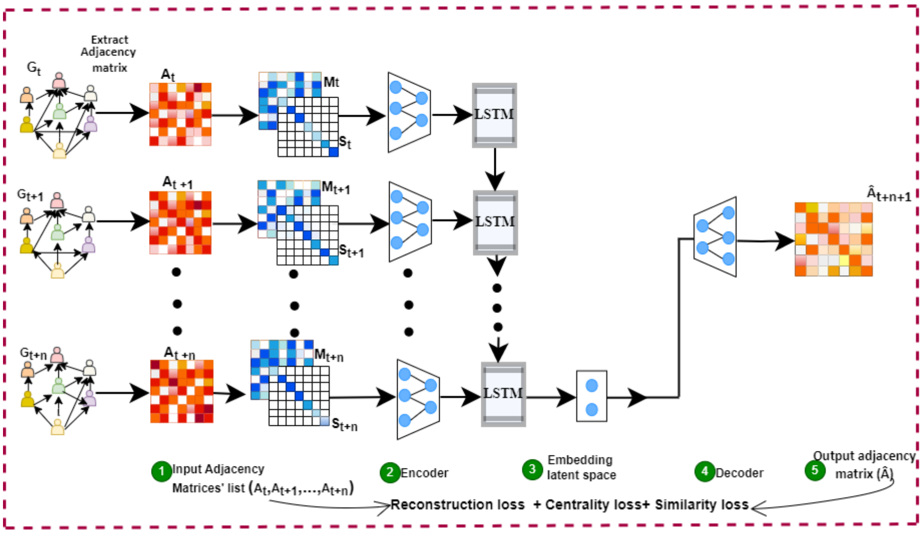


Figure 2: The Architecture of the proposed model. At each time step, an adjacency matrix is extracted from raw network data; then, the topological similarity and node strength centrality feature matrices are computed from the adjacency matrix following Eq.4 and Eq.9, respectively. These feature matrices are aggregated and fed to the fully connected auto-encoder, which learns the low-dimensional graph representation. The learned representation is then passed to the LSTM network to model the temporal information as guided by Eq.13. The LSTM output is then fed to the fully connected decoder. Finally, an adjacency matrix is sampled from the decoder output. The model loss Eq. 14 comprises the reconstruction loss, centrality loss, and topological similarity loss, all optimized simultaneously

Dataset	# Nodes	# Edges	# Snapshots
UCI	1899	59835	7
AS	6828	1947704	100
Math	24740	323357	77

Table 2: Description of the datasets

5.3 Evaluation Metrics

In the dynamic setting, we predict the presence of a link in the graph at time step $t+1$ based on the embeddings learned from all the previous graphs until time step t. Positive edge samples are generated by random sampling of real edges at time step t, while negative edge samples are obtained by sampling node pairs that are not connected by any edge. We used 5-fold cross-validation (5-fold-CV) to train the model, where data was randomly partitioned into 5 folds; four folds (80% of data) were used for training, and one fold (20% of data) was used for testing in each cross-validation run. The AUC (Area

Under the Receiver Operating Characteristic (ROC) Curve) score is used to evaluate the model's performance. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR). Where TPR is the proportion of edges that were correctly predicted to be positive out of all positive edges; it is computed as follows.

$$TPR = \frac{TP}{(TP + FN)} \tag{16}$$

Where TP (True positive) are the positive edges correctly predicted as positive, and FN (False Negative) are the negative edges incorrectly predicted as positive. Similarly, FPR is the proportion of edges that are incorrectly predicted to be positive out of all negative edges. It is computed as follows.

$$FPR = \frac{FP}{(TN + FP)} \tag{17}$$

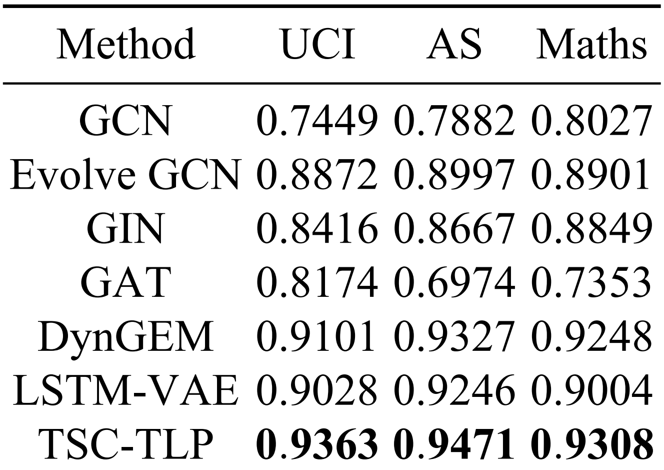
Where FP (True positive) are the negative edges incorrectly predicted as positive, and TN (True Negative) are the negative edges correctly predicted as negative. The AUC is equivalent to the probability of a randomly selected positive edge appearing above a randomly selected negative edge. It ranges from 0 to 1; the higher, the better. The mean AUC score is reported as the test accuracy.

5.4 Baselines

- GCN [Kipf and Welling, 2017]: Generalize classical Convolutional Neural Networks (CNN) to graph-structured data with a special layer-wise propagation rule.
- Evolve GCN [Pareja et al., 2020]: It is a dynamic variant of the GCN; it captures the dynamism of the graph by using an RNN to evolve the GCN parameters.
- SAGE [Hamilton et al., 2017]: It is a node embedding learning inductive framework where a function that generates embeddings is learned by sampling and aggregating features from a node's local neighborhood.
- GAT [Velickovic et al., 2018]: It is a variant of GCN that solves convolution and approximation-related problems by leveraging masked self-attentional layers to specify different weights to different nodes in a node's local neighborhood.
- DynGEM [Goyal et al., 2018]: It leverages deep autoencoders to learn embedding for dynamic graphs incrementally.
- VAE-LSTM: It is a variant of our proposed TSC-TLP, that utilizes variational autoencoder and LSTM layers to learn directly from the historical input graph adjacency matrices and predict the graph at the current time step.
- TSC-TLP-wc: This is a variant of our proposed TSC-TLP that has neither the topological similarity nor the node centrality preserving constraints imposed on the model learning.

5.5 Experimental results

5-fold cross-validation AUC link prediction results for the proposed model and the prominent baseline graph embedding studies on the 5 real-world networks are presented in Table 3. The best results are shown in bold. The Wilcoxon Signed Rank Test [Fix and Hodges Jr, 1955] is used to test the statistical significance of the results using a significance level of 0.05. In order to investigate the contribution of the similarity and strength-based features in the proposed TSC-TLP, we assess its variant VAE-LSTM that takes in the raw adjacency matrix as the input. We also verify the contribution of the topological similarity and strength centrality constraints on the model learning. We highlight the role of the temporal modules by assessing popular dynamic implementations (e.g., Evolve GCN) and its static variant GCN. Our proposed method TSCL-TLP is observed to significantly outperform the baseline models and its variant on all datasets due to its ability to not only capture the temporal network information but also to preserve the network topological similarity and centrality role of nodes during embedding learning. GIN is generally observed to be the second-best model due to its ability to learn certain simple graph structures that GCN-based variants, including Evolve GCN and GAT, may not learn. The static GCN is outperformed by its dynamic variant Evolve GCN signifying the capturing of temporal information during embedding learning. In order to examine the contributions of the specific modules in the proposed TSC-TLP we present its ablations. Table 4 compares the proposed TSC-TLP with its ablated versions. C-TLP is TSC-TLP ablation whose input is only centrality-based features, yet, TS-TLP is the TSC-TLP ablation that relies on only topological-similarity features as the input. TSC-TLP-wc is the TSC-TLP ablation whose input is both the centrality and topological similarity-based features as in TSC-TLP, though without the topological and centrality-based constraints imposed on the model learning.



Method	UCI	AS	Maths
GCN	0.7449	0.7882	0.8027
Evolve GCN	0.8872	0.8997	0.8901
GIN	0.8416	0.8667	0.8849
GAT	0.8174	0.6974	0.7353
DynGEM	0.9101	0.9327	0.9248
LSTM-VAE	0.9028	0.9246	0.9004
TSC-TLP	**0.9363**	**0.9471**	**0.9308**

Table 3: Comparison of the proposed TSC-TLP model and the baseline methods based on their average link prediction AUC scores on all datasets and for all the timestamps

Method	UCI	AS	Maths
C-TLP	0.7251	0.7880	0.7104
TS-TLP	0.9253	0.9361	0.9282
TSC-TLP-wc	0.9225	0.9299	**0.9316**
TSC-TLP	**0.9363**	**0.9471**	0.9308

Table 4: Comparison of the proposed TSC-TLP model and its ablated version based on their average link prediction AUC scores on all datasets and for all the timestamps

6 Discussion

This paper focuses on having an end-to-end unified model that learns network embeddings while preserving the topological structure, centrality role of nodes, and the underlying temporal graph information so as to predict graph links with high accuracy. We impose the topological similarity and node strength constraints on model learning to enforce preserving of the original topological structure and node strength centrality role in the learned embeddings.

6.1 Investigating the contribution of topological and the centrality node strength-based features in the proposed TSC-TLP model

The contribution of the topological and node strength-based features is clearly observed in Table 3 as the proposed TSC-TLP that utilizes both the topological and centrality-based features, is observed to outperform its variant VAE-LSTM whose input is the raw adjacent matrix. This highlights the role of the topology and node centrality features in characterizing the network during network embedding learning. As observed in Table 4, it is worth noting that deploying only topological similarity features yields better results as compared to centrality features. Yet, aggregating diverse features, including topological similarity and centrality-based features, results in more quality embeddings than using individual type features at a time. The topological and centrality-based features in the proposed TSC-TLP enhance capturing and preservation of the node relationships and the general graph connectivity information, thus boosting the quality of learned embeddings. This, in return, improves the accuracy of network applications, including link prediction that relies on the quality of the model embeddings.

6.2 Analysis of link prediction results based on the different topological similarity features deployed

In **Fig.** 3, the AUC scores based on features from the different popular topological similarity measures adopted, including common neighbor (CN), Jaccard Index (JI), Adamic-Ader (AA), and Salton Index (SI) are presented. As observed, generally, we obtained the best scores on using Adamic–Adar, followed by common neighbor. Thus, we adopted Adamic-Adar for the rest of our experiments, including results presented in Table 3.

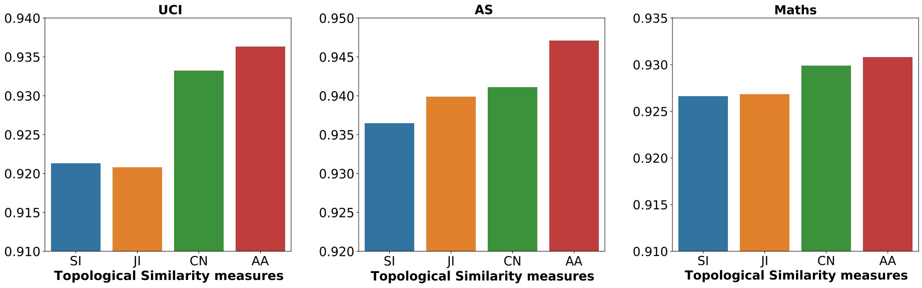


Figure 3: AUC scores based on the features extracted by the different topological similarity measures for the different datasets

6.3 Analyzing the impact of the network topological and node centrality preserving constraints on the embedding learning model

As observed in Table 4, on average, TSC-TLP is seen to outperform its ablated version TSC-TLP-wc which lacks topological and centrality-preserving constraints. This highlights the role of imposing such network structure preserving constraints on learning low dimensional representation of the network as discussed in other findings including [Shaw and Jebara, 2009, Sserwadda and Rekik, 2021, Bessadok et al., 2020]. These network property preserving constraints imposed on the model learning helped to further enforce the conserving of the network topological structure and node centrality role of the input graphs in the learned embeddings. Thus, improving the quality of learned embeddings and boosting the accuracy of the link prediction task that relies on the quality of the learned embeddings. However, for the Maths dataset in particular, the TSC-TLP-wc variant is seen to yield better results than the proposed TSC-TLP; this is likely due to the fact that the λ and γ hyper-parameters that control the weights attached to the topological similarity and node strength centrality constraints respectively were not fine-tuned. Thus, we did not achieve optimal results.

6.4 Investigating the influence of window size on link prediction accuracy

Here, we study the effect of change of n (i.e., the number of the temporal patterns learned in the recurrent layers) on the link prediction AUC. As a sample case, the AS dataset was chosen for this experiment. The number of temporal graphs used at a time is varied as t $\in$ [5,10,20,25,30] while noting the changes in the AUC. The test performance of models on the AS dataset with a varying number of temporal graphs is presented in **Fig.** 4. The AUC is observed to increase consistently with an increasing number of temporal graphs for dynamic methods, including TSC-TLP, DynGEM, and EvolveGCN, as such models take advantage of the memory units to store and learn from previous temporal patterns.

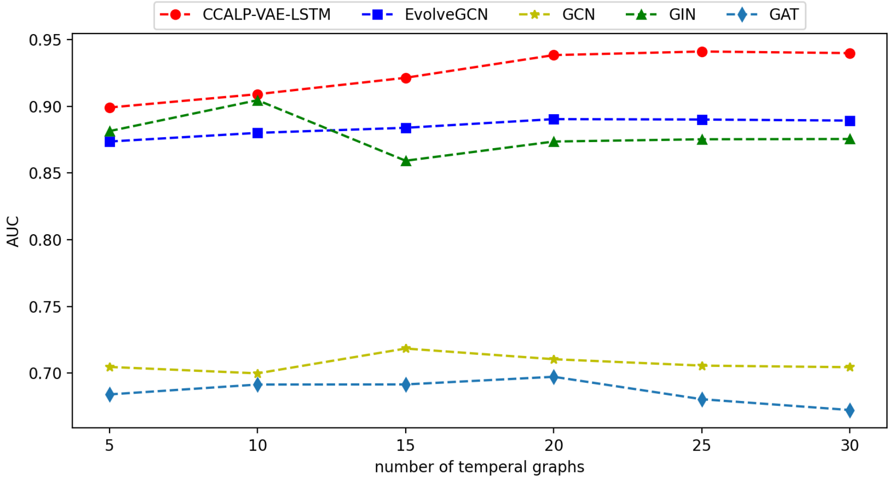


Figure 4: AUC vs. Increasing number of temporal graphs on AS dataset

However, beyond n=25, generally, no significant improvement in AUC is noted. It is interesting to note that for static methods like GCN, initially, the AUC increases with the window size, but after a while (e.g., beyond window size= 15 for GCN), the AUC is observed to decrease since such static methods lack memory units to store and learn from long previous temporal patterns.

6.5 Examining models' efficiencies in preserving the node centrality role in their predicted graphs

To understand the models' abilities to preserve the structural centrality roles of nodes in the learned embeddings, we compute the node centralities of nodes in ground truth test graphs and the centralities of the same nodes in predicted graphs. The Average MSE of the ground truth and predicted graphs centrality scores are shown in **Fig.** 5.

We examined the graphs on three important centrality measures, including eigenvector centrality [Wasserman and Faust, 1994], a global variant of degree centrality for measuring the node's influence in a network, closeness centrality [Bavelas, 1950] that tells how fast the information flow will be in the network based on the reference node and betweenness centrality [Freeman, 1977] that quantifies the individual node's role in bridging between nodes in a network. As observed in **Fig.** 5, the proposed TSC-TPL records the minimum MSE on both datasets for all centrality measures apart from a single case of betweenness centrality prediction on the $Maths$ dataset where DynGEM performs slightly better. Numerically, TSC-TLP records 30%, 59%, and 12% reductions in MSE on the UCI, AS, and $Maths$ datasets, respectively (an average reduction of 37%) over the best benchmark (DynGEM). This highlights the efficiency of the proposed TSC-TLP in preserving the structural centrality role of nodes in a network. Thanks to the topology and centrality-based features and the respective network property constraints imposed on the model learning.

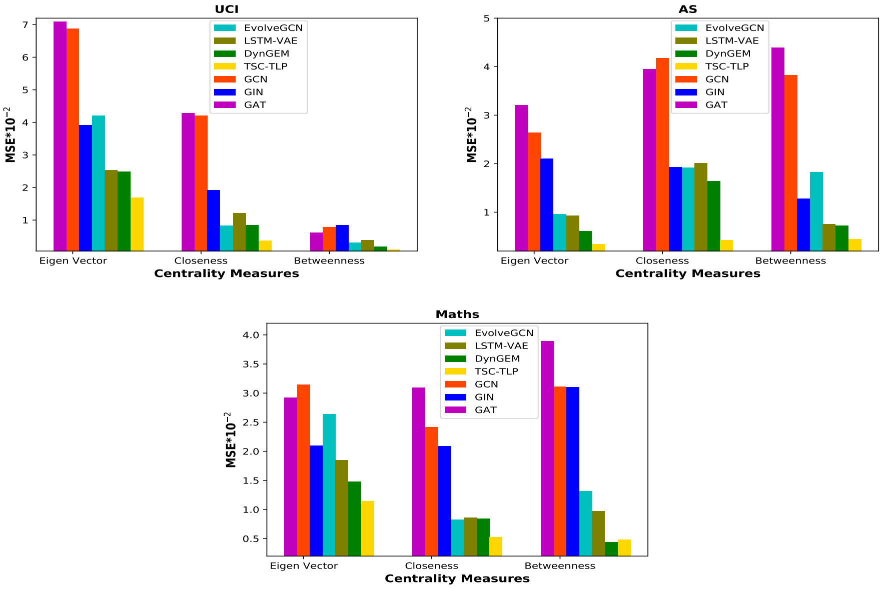


Figure 5: MSE in predicting centralities of graphs predicted by the different methods on all datasets

6.6 Limitations and future work

The major limitations of our work include the following. First, we adopted a few local topological similarity measures. In future work, a comprehensive study involving other topological similarity measures, including Hub Depressed Index [Ravasz et al., 2002], Resource Allocation Index [Zhou et al., 2009], and Preferential Attachment [Barabási and Albert, 1999] will be our target. Second, vital hyper-parameters, including λ and γ, that regulate the weights assigned to centrality and topological constraints were not fine-tuned; hence our results are not optimal. Automatic learning and fine-tuning of these hyperparameters would yield better results. Lastly, we aimed to preserve the topological and centrality network information. Other network properties, including the community and hierarchical structures, could be examined to improve the embedding quality.

7 Conclusion

In this work, an end-to-end topological similarity and centrality aware link prediction for temporal networks model (TSC-TLP) is proposed that preserves not only the topological similarity and centrality information in the learned embeddings but also explores the underlying temporal information in the evolving networks. The model utilizes both topological similarity and structural centrality features and the relevant constraints imposed on its learning to respectively capture and enforce the preservation of the network topology and the structural centrality role of nodes in learned embeddings. Thus, yielding high-quality embeddings that boost the link prediction accuracy. The proposed TSC-TLP model outcompetes the best benchmark by an average of 4% improvement in link prediction accuracy and 37% reduction in MSE on centrality prediction. So far, we have explored only local topological similarity-based features. Notably, our framework can be extended to capturing features based on global topological similarity measures, including; Matrix Forest Index [Chebotarev and Shamis, 1997] and the Leicht-Holme-Newman Index [Leicht et al., 2006].

Acknowledgements

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[Adamic and Adar, 2003] Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. *Social networks*, 25(3):211–230.

[Aghdam and Zanjani, 2021] Aghdam, M. H. and Zanjani, M. D. (2021). A novel regularized asymmetric non-negative matrix factorization for text clustering. *Information Processing & Management*, 58(6):102694.

[Aleskerov and Shvydun, 2018] Aleskerov, F. and Shvydun, S. (2018). Stability and similarity in networks based on topology and nodes importance. In *International Conference on Complex Networks and their Applications*, pages 94–103. Springer.

[Barabási and Albert, 1999] Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *science*, 286(5439):509–512.

[Bavelas, 1950] Bavelas, A. (1950). Communication patterns in task-oriented groups. *The journal of the acoustical society of America*, 22(6):725–730.

[Bessadok et al., 2020] Bessadok, A., Mahjoub, M. A., and Rekik, I. (2020). Topology-aware generative adversarial network for joint prediction of multiple brain graphs from a single brain graph. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 551–561. Springer.

[Bhagat et al., 2011] Bhagat, S., Cormode, G., and Muthukrishnan, S. (2011). *Node Classification in Social Networks*. Springer US, Boston, MA.

[Bucur and Holme, 2020] Bucur, D. and Holme, P. (2020). Beyond ranking nodes: Predicting epidemic outbreak sizes by network centralities. *PLOS Computational Biology*, 16(7):e1008052.

[Chebotarev and Shamis, 1997] Chebotarev, P. and Shamis, E. (1997). The matrix-forest theorem and measuring relations in small social groups. *Autom.Remote Control*, 58(9).

[Chen et al., 2018] Chen, H., Perozzi, B., Hu, Y., and Skiena, S. (2018). Harp: Hierarchical representation learning for networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[Chen et al., 2021] Chen, J., Wang, X., and Xu, X. (2021). Gc-lstm: graph convolution embedded lstm for dynamic network link prediction. *Applied Intelligence*, pages 1–16.

[Chen et al., 2019] Chen, J., Zhang, J., Xu, X., Fu, C., Zhang, D., Zhang, Q., and Xuan, Q. (2019). E-lstm-d: A deep learning framework for dynamic network link prediction. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

[Coşkun and Koyutürk, 2021] Coşkun, M. and Koyutürk, M. (2021). Node similarity-based graph convolution for link prediction in biological networks. *Bioinformatics*, 37(23):4501–4508.

[Dallas et al., 2019] Dallas, T. A., Carlson, C. J., and Poisot, T. (2019). Testing predictability of disease outbreaks with a simple model of pathogen biogeography. *Royal Society open science*, 6(11):190883.

[Fisher, 1936] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

[Fix and Hodges Jr, 1955] Fix, E. and Hodges Jr, J. (1955). Significance probabilities of the wilcoxon test. *The Annals of Mathematical Statistics*, pages 301–312.

[Freeman, 1977] Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41.

[Freeman et al., 1991] Freeman, L. C., Borgatti, S. P., and White, D. R. (1991). Centrality in valued graphs: A measure of betweenness based on network flow. *Social networks*, 13(2):141–154.

[Geethanjali, 2015] Geethanjali, P. (2015). *Fundamentals of Brain Signals and Its Medical Application Using Data Analysis Techniques*. Springer.

[Goyal et al., 2020] Goyal, P., Chhetri, S. R., and Canedo, A. M. (2020). dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems*, 187:104816.

[Goyal et al., 2018] Goyal, P., Kamra, N., He, X., and Liu, Y. (2018). Dyngem: Deep embedding method for dynamic graphs. *3rd International Workshop on Representation Learning for Graphs (ReLiG)*, 1805.

[Hamilton et al., 2017] Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035.

[Huang and Zeng, 2006] Huang, Z. and Zeng, D. D. (2006). A link prediction approach to anomalous email detection. In *2006 IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1131–1136. IEEE.

[Jaccard, 1912] Jaccard, P. (1912). The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, volume 3.

[Kipf and Welling, 2017] Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, pages 1–14.

[Klein, 2010] Klein, D. (2010). Centrality measure in graphs. *Journal of mathematical chemistry*, 47(4):1209–1223.

[Koutrouli et al., 2020] Koutrouli, M., Karatzas, E., Paez-Espino, D., and Pavlopoulos, G. A. (2020). A guide to conquer the biological network era using graph theory. *Frontiers in bioengineering and biotechnology*, 8:34.

[Kwon et al., 2019] Kwon, H., Choi, Y.-H., and Lee, J.-M. (2019). A physarum centrality measure of the human brain network. *Scientific reports*, 9(1):1–13.

[Leicht et al., 2006] Leicht, E. A., Holme, P., and Newman, M. E. (2006). Vertex similarity in networks. *Physical Review E*, 73(2):026120.

[Li et al., 2014] Li, J., Zhang, L., Meng, F., and Li, F. (2014). Recommendation algorithm based on link prediction and domain knowledge in retail transactions. *Procedia Computer Science*, 31:875–881.

[Li et al., 2018] Li, S., Huang, J., Zhang, Z., Liu, J., Huang, T., and Chen, H. (2018). Similarity-based future common neighbors model for link prediction in complex networks. *Scientific reports*, 8(1):1–11.

[Liu et al., 2019] Liu, W., Li, X., Liu, T., and Liu, B. (2019). Approximating betweenness centrality to identify key nodes in a weighted urban complex transportation network. *Journal of Advanced Transportation*, 2019.

[Ma et al., 2020] Ma, A., You, F., Jing, M., Li, J., and Lu, K. (2020). Multi-source domain adaptation with graph embedding and adaptive label prediction. *Information Processing & Management*, 57(6):102367.

[Mallek, 2018] Mallek, S. (2018). *Social Network Analysis: Link prediction under the Belief Function Framework*. PhD thesis, Artois.

[Monti et al., 2017] Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., and Bronstein, M. M. (2017). Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124.

[NarasingaRao et al., 2018] NarasingaRao, M., Venkatesh Prasad, V., Sai Teja, P., Zindavali, M., and Phanindra Reddy, O. (2018). A survey on prevention of overfitting in convolution neural networks using machine learning techniques. *International Journal of Engineering and Technology (UAE)*, 7(2.32):177–180.

[Newman, 2001] Newman, M. E. (2001). Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102.

[Newman, 2004] Newman, M. E. (2004). Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133.

[Newman and Girvan, 2004] Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.

[Nie et al., 2017] Nie, F., Zhu, W., and Li, X. (2017). Unsupervised large graph embedding. In *Thirty-first AAAI conference on artificial intelligence*, page 2422–2428.

[Nobre et al., 2022] Nobre, G. P., Ferreira, C. H., and Almeida, J. M. (2022). A hierarchical network-oriented analysis of user participation in misinformation spread on whatsapp. *Information Processing & Management*, 59(1):102757.

[Pareja et al., 2020] Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., and Leiserson, C. (2020). Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5363–5370.

[Perozzi et al., 2014] Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

[Qiao and Hu, 2020] Qiao, C. and Hu, X. (2020). A neural knowledge graph evaluator: Combining structural and semantic evidence of knowledge graphs for predicting supportive knowledge in scientific qa. *Information Processing & Management*, 57(6):102309.

[Ravasz et al., 2002] Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabási, A.-L. (2002). Hierarchical organization of modularity in metabolic networks. *science*, 297(5586):1551–1555.

[Sajjad et al., 2020] Sajjad, H. P., Docherty, A., and Tyshetskiy, Y. (2020). Efficient representation learning using random walks for dynamic graphs. *Applied Network Science*, 5(8).

[Salton and McGill, 1983] Salton, G. and McGill, M. J. (1983). *Introduction to modern information retrieval*. mcgraw-hill.

[Shaw and Jebara, 2009] Shaw, B. and Jebara, T. (2009). Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 937–944.

[Sserwadda and Rekik, 2021] Sserwadda, A. and Rekik, I. (2021). Topology-guided cyclic brain connectivity generation using geometric deep learning. *Journal of Neuroscience Methods*, 353:108988.

[Velickovic et al., 2018] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, volume 1710.10903.

[Wang et al., 2016] Wang, D., Cui, P., and Zhu, W. (2016). Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234.

[Wang et al., 2010] Wang, Z., Tan, Y., and Zhang, M. (2010). Graph-based recommendation on social networks. In *2010 12th International Asia-Pacific Web Conference*, pages 116–122. IEEE.

[Wasserman and Faust, 1994] Wasserman, S. and Faust, K. (1994). *Social network analysis: Methods and applications*. Cambridge university press.

[Wu et al., 2020] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

[Yan et al., 2006] Yan, S., Xu, D., Zhang, B., Zhang, H.-J., Yang, Q., and Lin, S. (2006). Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE transactions on pattern analysis and machine intelligence*, 29(1):40–51.

[Zhou et al., 2020] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.

[Zhou et al., 2009] Zhou, T., Lü, L., and Zhang, Y.-C. (2009). Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630.

[Zhu et al., 2016a] Zhu, L., Guo, D., Yin, J., Steeg, G., and Galstyan, A. (2016a). Scalable temporal latent space inference for link prediction in dynamic social networks. *IEEE Transactions on Knowledge & Data Engineering*, 28(10):2765–2777.

[Zhu et al., 2016b] Zhu, L., Guo, D., Yin, J., Steeg, G. V., and Galstyan, A. (2016b). Scalable temporal latent space inference for link prediction in dynamic social networks. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2765–2777.