Link Prediction by Analyzing Temporal Behavior of Vertices

Kalyani Selvarajah[1][0000−0001−8551−0648], Ziad Kobti[1][0000−0001−9503−9730], and Mehdi Kargar[2][0000−0001−6937−3508]

[1] School of Computer Science, University of Windsor, Windsor, ON, Canada
{selva111,kobti}@uwindsor.ca
[2] Ted Rogers School of Management, Ryerson University, Toronto, ON, Canada
kargar@ryerson.ca

Abstract. Complexity and dynamics are challenging properties of real-world social networks. Link prediction in dynamic social networks is an essential problem in social network analysis. Although different methods have been proposed to enhance the performance of link prediction, these methods need significant improvement in accuracy. In this study, we focus on the temporal behavior of social networks to predict potential future interactions. We examine the evolving pattern of vertices of a given network $\mathcal{G}$ over time. We introduce a time-varying score function to evaluate the activeness of vertices that uses the number of new interactions and the number of frequent interactions with existing connections. To consider the impact of timestamps of the interactions, the score function engages a time difference of the current time and the time of the interaction occurred. Many existing studies ignored the weight of the link in the given network $\mathcal{G}$, which brings the time-varied details of the links. We consider two additional objective functions in our model: a weighted shortest distance between any two nodes and a weighted common neighbor index. We used Multi-Layer Perceptron (MLP), a deep learning architecture as a classifier to predict the link formation in the future and define our model as a binary classification problem. To evaluate our model, we train and test with six real-world dynamic networks and compare it with state-of-the-art methods as well as classic methods. The results confirm that our proposed method outperforms most of the state-of-the-art methods.

Keywords: Link Prediction · Multilayer Perceptron · Dynamic Networks · Social Network Analysis

1 Introduction

Social Networks (SN) can be used to model a comprehensive range of real-life phenomena and examine the world around us. It ranges from online social interaction, including Facebook, Twitter, and LinkedIn to human interactions such as co-authorship, healthcare, and terrorist networks. Social networks analysis is the study of such networks to discover common structural patterns and explains

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_19

their emergence through computational models of network formation. The complexity and dynamics are essential properties of real-world social networks. Since these networks evolve quickly over time through the appearance or disappearance of new links and nodes, the connection becomes stronger and weaker, and underlying network structure changes with time. Therefore it has become high challenges for researchers in order to examine various research issues in social network analysis such as classification of a node, detecting the communities, formation of teams, and predicting links between nodes.

Understanding the mechanism of how the networks change over time is a crucial problem that is still not well understood [13]. Significant efforts have been made to explain the evolution of networks during the past decades [5]. However, such researches are yet to achieve the desired results, leaving the door open for further advances in the field. Throughout the last decades, analyzing temporal networks has received much attention among researchers as it has enormous applications in different disciplines such as co-authorship [2], the recommendation of friends [30] and website links. Recently, dynamic link predictions have been approached by various mechanisms and achieved promising results. However, the features of networks vary from each other, and the existing studies are not efficient to represent the importance of nodes and links. The objective of this paper is to address these issues and examines the dynamic nature of social networks.

Link prediction problem needs to be solved by determining the potentialities of the appearance or disappearance of links between all node pairs of the given network [28]. However, for example, in a collaborative network, if two experts (*i.e.*, vertices) collaborate once on any project, their link remains permanent, although any one of them stops interacting with the others further. Therefore, their link become weak in the future, while experts who have frequent interactions their link become strong. In this regard, we observe the behavior of individuals in the collaborative network when they need to decide which new collaborations might prove fruitful in addition to existing connections. Before anyone connects with others in the network, they usually examine several factors, including whether the people are active throughout the past or not, and are they working on similar projects that they have skills. Therefore, our study takes these factors into account and propose a new model, LATB, to predict the links which occur with others in the future. In this paper, we propose a model for link prediction problem on dynamic social networks and make the following major contributions.

1. Active individuals in social networks are popular among both existing members and new members who like to join the network. They believe that active individuals, for instance, in the co-authorship network, always update their research with current trends as well as being open to new ideas. So, to evaluate the activeness of any member, we consider two factors on the temporal network. (a) The score for constructing new connections (b) The score for the increased number of interactions with existing connections (How much the existing link becomes strong). We introduce a new score function to incorporate the impact of the timestamps and the gap between the current

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_19

time and the time of the interaction occurred. Besides this, we introduce a probability function based on the activeness score of a pair of nodes to decide the likelihood of occurring a new link.
2. The smaller distance between any two individuals is higher the chances of future interaction. We incorporate the weighted shortest distance in LATB. In addition to this, we include another objective function, the weighted common neighbor index, which incorporates the time to evaluate the changes of strength of the neighbors' relationship.
3. In LATB, we used Multi-Layer Perceptron (MLP) as a classifier to predict the link formation in the future and defined our model as a binary classification problem.

The remaining of the paper is organized as follows. Section 2 describes related works. Section 3 specifies the problem definitions. Section 4 presents the experimental setup and the corresponding results. Finally, Section 6 concludes the research idea of this paper with directions for future work.

2 Related Works

Link prediction problem on the static network examines a single snapshot of a network structure at time t as an input, and then predicts possible unobserved links at time $t'(t \leq t')$ [13]. On the other hand, link prediction in dynamic networks investigates the evolution of networks over time as a sequence of snapshots and then predicts new links in the future. This section presents an overview of the link prediction problems on social networks. Several methods have been proposed to deal with the link prediction problem on the temporal network systems during the past decade.

The researchers designed a lot of topology-based similarity metrics for link prediction such as Common Neighbors (CN) [16], Adamic-Adar Coefficient (AA) [1], and Katz (KZ)[9]. Since the weights of links are rarely taken into account, many researchers modified those metrics in order to adopt the dynamic features of the social networks. The authors [15] examine the link prediction based on connection weight score structural properties of a given network. Zhu et al. [32] proposed a weighted mutual information model which is to estimate the effect of network structures on the connection likelihood by considering the benefits of both structural properties and link weights.

Potgieter et al. [21] showed that temporal metrics are valuable features in terms of accuracy. Tylenda et al. [25] proposed a graph-based link prediction algorithm and integrated it with temporal information and extended the local probabilistic model to involve time awareness. Yao et al. [31] used time-decay to manage the weight of the links and modified the common neighbor index to includes nodes in 2-hop. The authors [10] presented a time frame based unsupervised link prediction method for directed and weighted networks and derived a score for potential links in a time-weighted manner.

Tong W et al. [29] examined the concepts of the temporal trend of nodes by considering the changes of the degree over time using the structural perturbation method. Munasinge et al. [14] studied the impact of a relationship between timestamps of interactions and strength of the link for the future.

Xiaoyi Li et al. [12] proposed a deep learning method, conditional temporal restricted Boltzmann machine, which adopted a combination of feature engineering and CNN to predicts links. Recently, Goyal et al. [6] proposed DynGEM, which uses the recent advances in deep learning methods, autoencoders for graph embeddings to handle growing, dynamic graphs and for link prediction. Wang et al. [27] examined relational deep learning to jointly model high-dimensional node attributes and link structures with layers of latent variables and proposed generalized variational inference algorithm for learning the variables and predicting the links.

3 A Model for Dynamic Link Prediction

3.1 Problem Definition

A dynamic network is evolving over time and can be considered as a sequence of network snapshots within a time interval. The size of the network can occasionally shrink or expand as the network evolves. In this work, we focus on undirected weighted graphs.

Given a series of snapshots $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{t-1}\}$ of an evolving graph $\mathcal{G}_T = \langle \mathcal{V}, \mathcal{E}_T \rangle$, where the edge $e = (u, v) \in \mathcal{E}_{t'}$ represents a link between $u \in \mathcal{V}_{t'}$ and $v \in \mathcal{V}_{t'}$ at a particular time t'. The dynamic link prediction approaches attempt to predict the likelihoods of links in the next time step $\mathcal{G}_t$. The list of graphs $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{t-1}\}$ corresponding to a list of symmetric adjacency matrices $\{A_1, A_2, \ldots, A_{t-1}\}$. The adjacency matrix A_T of $\mathcal{G}_T$ is a $\mathcal{N} \times \mathcal{N}$ matrix where each element $A_T(i, j)$ takes 1 if the nodes $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$, are connected at least once within time period T and takes 0 if they are not. Given a sequence of $(t-1)$ snapshots $\{A_1, A_2, \ldots, A_{t-1}\}$, the goal is to predict the adjacency matrix A_t at future time t.

3.2 Node Activeness

The idea of node activeness is highly related to the temporal behaviors of nodes. We can determine the active nodes through the analysis of the time-varying historical information of the nodes. To decide the activeness of the nodes, we can examine how they interact with others (nodes) throughout the timeframe. With any temporal network involving humans, we believe that the following factors are highly relevant to decide the activeness of nodes.

New Connections: A node can remain inactive or active. It can be decided based on how often a node made the new interactions throughout the time frame. Let us consider any two members A and B of a given dynamic network $\mathcal{G}$ with the same number of new connections in the past; A might be connected at an early stage and not creating any new connection later (can be named as sleeping node), while node B formed most of his or her connection at the later stage (active node). The node B would attract more new people and have a high probability of generating new connections in the near future. We believe that considering this behavior of the node is significant in predicting the new links.

Definition 1. ***(Score for New Connections)*** *Given a network $\mathcal{G}_T\langle\mathcal{V},\mathcal{E}_T\rangle$ and a set of nodes $A = \langle a_1, a_2, \ldots, a_n\rangle$ at time t_m. Let's say the time windows to estimate the new connections are $|t_m^1 - t_m^2|$,$|t_m^2 - t_m^3|$ $\ldots|t_m^{n-1} - t_m^n|$, where $t_m^1, t_m^2 \ldots t_m^n$ are consecutive time stamps. The score of building new connection at time t_m^n of a member or node of the network a_k can be defined as:*

$$\mathcal{SN}(a_k) = \sum_{i=1}^{t_m^n} \frac{\mathcal{NC}_{a_k}^{t_i}}{|t_m^n - t_i| + 1} \tag{1}$$

where $\mathcal{NC}_{a_k}^{t_i}$ is the number of new connection made by the node a_k at time t_i. The term $|t_m^n - t_i|$ is the timestamp between current time and the selected time that the number of new connections has been made by the node a_k. If the difference between t_m^n and t_i is high, the value of $\mathcal{NC}_{a_k}^{t_i}$ over $|t_m^n - t_i| + 1$ become smaller although a node made more number of new connections at early time ($\mathcal{NC}_{a_k}^{t_i}$ is large). However, this is opposite for the nodes which made new connection recently. The timestamp has an addition of one to avoid the denominator become infinite when the two timestamps are equal.

The Frequency of Interaction: In addition to the new interactions, a node can continuously associate with the existing connected nodes. It is another way to decide the activeness of the node. Let us consider two nodes A and B with the same number of existing connections in the past; A might have several interactions at an early stage and not interacting with them later, while B is having frequent interactions with existing connections throughout the time frame. The later one (B) would attract more new nodes and have a high probability of generating new connections in the near future.

Definition 2. ***(Score for Frequent Interactions)*** *The score of frequent collaborations with existing connection at time t_m^n of a node a_k can be defined as:*

$$\mathcal{SE}(a_k) = \sum_{i=1}^{t_m^n} \frac{\mathcal{EC}_{a_k}^{t_i}}{|t_m^n - t_i| + 1} \tag{2}$$

where $\mathcal{EC}_{a_k}^{t_i}$ is the number of frequent collaboration with existing connection by a node a_k at time t_i.

For instance, let's consider a network of authors (nodes) at certain year (2012) and the number new connection which both A and C made through last seven years (till 2019) can be shown as Figure 1.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_19

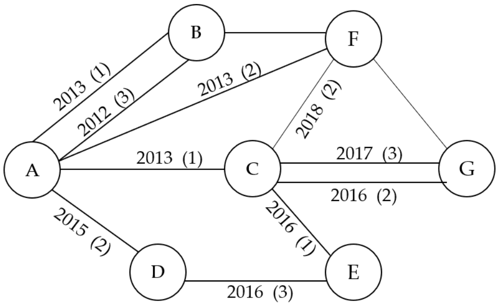

Fig. 1: Interaction with new and existing nodes throughout the time

$$
\begin{aligned}
\mathcal{SN}(A) &= \sum_{i=2012}^{2019} \frac{\mathcal{NC}_A^i}{|2019 - i| + 1} \\
&= \frac{3}{|2019 - 2012| + 1} + \frac{4}{|2019 - 2013| + 1} + \frac{2}{|2019 - 2015| + 1} \\
&= 1.382
\end{aligned}
$$

Similarly, the value of $\mathcal{SN}(C)$ can be evaluated to 2.89 as the information given in Figure 1. Although both A and C made the equal number of new connections ($= 9$) in the past seven years, the score has a huge difference. The reason is C built new connections recently than A. This shows that nodes, which generate new connections recently have more attractive than others. Likewise, we can evaluate the score value of active nodes in terms of frequent interaction with existing connections.

Since both building new interactions and frequent interactions with existing connections have an influence on deciding the active node, the combination of these scores is a proper way to determine the score of the active node as given in equation 3.

$$
\mathcal{SA}(a_k) = \lambda \sum_{i=1}^{t_m^n} \frac{\mathcal{NC}_{a_k}^{t_i}}{|t_m^n - t_i| + 1} + (1 - \lambda) \sum_{i=1}^{t_m^n} \frac{\mathcal{EC}_{a_k}^{t_i}}{|t_m^n - t_i| + 1} \qquad (3)
$$

where λ is a tradeoff value between the score for building new connections and expanding existing connections. Algorithm 1 describe the step by step process of calculating node activeness score for all nodes and store it in a lookup table.

At this point, every node is assigned by a score based on their activity on dynamic networks. However, to maintain the range of values between 0 and 1, we normalize each score. Inspired by configuration model, the probability $\mathcal{P}_{a_i,a_j}$ of a link exists between any two nodes a_i and a_j would be proportional to $\mathcal{SA}(a_i).\mathcal{SA}(a_j)$. Since the probability should be between 0 and 1, we drive the equation for this value by multiplying the reciprocal of the total activeness of

Algorithm 1 Node Activeness

Input: Current Time stamp t_c, Snapshots of a given network G_T, List of vertices L_V, Trade off value λ
Output: Activeness Score lookup table $\mathcal{SA}$

```
$\mathcal{SA} \leftarrow \{\}$
$T[\,] \leftarrow \{t_1, t_2, \ldots t_k\}$ time steps
for all $n \in L_V$ do
    $Ex \leftarrow 0$ Existing Connection Score
    $Nw \leftarrow 0$ New Connection Score
    $nb \leftarrow \Gamma\,[G_{t_0}(n)]$ collaborated nodes at time $t = 0$
    for all $t_i$ in $T$ do
        $nb_i[\,] \leftarrow \Gamma\,[G_{t_i}(n)]$
        for all $m \in nb$ do
            if $m \in nb_0$ then
                $Ex \leftarrow Ex + \frac{w_{G_{t_i}}(m,n)}{t_c - t_i + 1}$
            else
                $Nw \leftarrow Nw + \frac{w_{G_{t_i}}(m,n)}{t_c - t_i + 1}$
            end if
        end for
        $nb \leftarrow nb \cup nb_i$
    end for
    $P \leftarrow \lambda.Ex + (1 - \lambda)Nw$
    $\mathcal{SA} \leftarrow \{$n: $P\}$
end for
return $\mathcal{SA}$
```

the nodes in the networks as given in equation 5.

$$\mathcal{P}_{a_i,a_j} = \frac{\mathcal{SA}(a_i)\mathcal{SA}(a_j)}{\sum_{k=1}^{n} \mathcal{SA}(a_k)} \tag{4}$$

where $\mathcal{P}_{a_i,a_j}$ is the probability of existing link between node a_i and a_j, $\mathcal{SA}(a_i)$ and $\mathcal{SA}(a_j)$ are the popularity scores of nodes a_i and a_j respectively and n is the total number of nodes in the networks.

We name our proposed method LATB (Link prediction by Analyzing Temporal Behaviour of vertices), because it highlights the behaviors of vertices.

3.3 Similarity Metrics

In the past, the majority of researches [13, 19] have examined the accuracy of several heuristics for link prediction such as Adamic Adar, Preferential Attachment, and Jaccard Coefficient. In this research, we consider two modified forms of heuristics: weighted shortest path distance and weighted common neighbors.

Weighted Shortest Path Distance: In the undirected social network G, if some nodes have past interaction, their associated nodes in G are connected by an edge. If many levels of past interactions between two nodes are taken into account, then the input graph G is weighted. In this case, the smaller the edge weight between two nodes, the two nodes had more interactions in the past and have higher chances of interactions in the future. The distance between two nodes a_i and a_j, specified as $dist(a_i, a_j)$, is equal to the sum of the weights on the shortest path between them in the input graph G. If a_i and a_j are not connected in graph G, *i.e.*, there is no path between a_i and a_j in G, the distance between them is set to ∞.

$$\mathcal{SD}_{a_i,a_j} = wdist(a_i, a_j) \tag{5}$$

Weighted Common Neighbors: The Common Neighbors (CN) is the most widely used index in link prediction and evidence to the network transitivity property. It counts the number of common neighbors between node pair a_i and a_j. Newman et al. [16] has estimated this quantity in the context of collaboration networks. The probability that a_i and a_j collaborate in the future can be written as 6.

$$\mathcal{CN}_{a_i,a_j} = |\varGamma(a_i) \cup \varGamma(a_j)| \tag{6}$$

where $\varGamma(a_i)$ and $\varGamma(a_j)$ consists of number of neighbors of the node a_i and a_j in $\mathcal{G}$ respectively.

As mentioned in the above definition, the common neighbors only consider the binary relations between nodes and ignore the time-varying nature and number of link occurrences. We adopt the time-varied weights into the common neighbors, which can give better predictions [8].

$$\mathcal{CN}^{tw}_{a_i,a_j} = \sum_{|\varGamma(a_i) \cup \varGamma(a_j)|} \mathcal{W}^t(a_i, a_k) + \mathcal{W}^t(a_k, a_j) \tag{7}$$

where $\mathcal{W}^t(a_i, a_k) = \mathcal{W}(a_i, a_k) - \beta(t' - t)$, $\mathcal{W}(a_i, a_k)$ is original weight at time t, β is an attenuation factor and t' is the time considered for prediction.

3.4 Multilayer Perceptron (MLP) Framework

We treat the link prediction problem as a binary classification problem. We used MLP as a classifier. In this regards, we generate a dataset for all existing links in a last time step of a given dynamic network $\mathcal{G}_\mathcal{T}$, for the positive link class, where for any two vertices i and j, the link between i and j, $ij \in E_t$. We generate negative link class, where any two vertices i and j, $ij \notin E_t$ by using downsampling technique to avoid imbalance problem. We assign the binary cross-entropy as loss function, which can be written as:

$$BCE = -y.log(p) - (1 - y)log(1 - p) \tag{8}$$

where y is binary indicator (0 or 1), p is predicted probability.

Finally, we build and train a neural network for link prediction. MLP is one of the most common and a variant of the original Perceptron model [22]. Here,

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_19

we only briefly discuss the components of an MLP since this paper is not about MLP innovations. A typical MLP system can be built with layers of neurons, as shown in Figure 2. Each neuron in a layer calculates the sum of its inputs (x) that are carried through an activation function (f). The output (O) from the network can be written as:

$$\mathcal{O}_{jk} = F_k(\sum_{i=1}^{N_{k-1}} w_{ijk} x_{i(k-1)} + \beta_{jk}) \tag{9}$$

where $\mathcal{O}_{jk}$ is the neuron j^{th} output at k^{th} layer and β_{jk} is bias weight for neuron j in layer k, respectively.

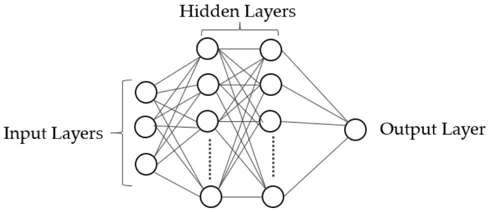

Fig. 2: MLP neural network

4 Experimental Results

We conduct extensive experiments to test our model with five real-world dynamic networks and use AUC (Area Under Curve) as evaluation metrics.

4.1 Dataset

We use six real world dynamic networks: **Enron** corpus [11] and **Radoslaw** [23] are email communication networks. Each node specifies an employee and link represents email conversation among employees. Enron has the details from 6 January 1998 until 4 February 2004 while Randoslaw is from January 2^{nd} 2010 to September 30^{th} 2010. **Contact** [4] is data from wireless devices carried by people. Every node is people, and a link established when they contacted. The contact list represents the active contacts during 20-second intervals of the data collection. **College Messages** [17] have private messages sent on an online social network at the University among college people. **EU-core** [18] is an email data from a large European research institution. Link is the communication between members from 4 different departments. **Mathoverflow** [18] has the interactions on the stack exchange web site Math Overflow.

In the beginning, we sort the dataset in ascending order of time, and then we process a sequence of snapshots for each dataset at a fixed interval. We split every dynamic networks into five time frames G_1, G_2, G_3, G_4, G_5. We evaluate each scoring value at the last snapshot.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_19

Dataset	$\lvert V\rvert$	$\lvert E_T\rvert$	Time span in days
Enron	151	50571	165
Radoslaw	167	82900	272
Contact	274	28200	4
CollegeMessages	1899	59835	193
EU-core	986	332334	803
Mathoverflow	24818	506550	2350

Table 1: The statistical information of each real-world dynamic networks.

4.2 Experimental Setup

We implement our model in Python3, processed the dataset on IBM cluster, the specification of POWER8 52 processor 256 GB of RAM. We trained and tested our model in an Nvidia GTX 1050Ti, 4 GB GPU with 768 CUDA cores.

In the weighted common neighbor index, we set the attenuation factor β to 0.001. To evaluate the active score of the nodes, we assign tradeoff factor λ to 0.5, because we believe that both the score for new connections and the score for frequent interaction with existing connections are equally important to decide the activeness of a person.

In the MLP, the first layer has four neurons with the ReLu activation function. We use two hidden layers of 32 neurons. The output layer contains a single neuron with the Sigmoid activation function. We train the neural network for 100 epochs. We use 80% training set, 10% validation set, and 10% testing set. We repeat the above process for ten times and find the average AUC.

4.3 Baseline Methods

We use various methods as the baselines, including classical methods such as Common Neighbors (CN), Jaccard coefficient (JC), Adamic Adar (AA) and Preferential attachment (PA), and network embedding methods such as node2vec, LINE, DeepWalk, and SDNE. The brief introduction of these methods is listed as follow:

- Common Neighbors (CN)[16]: It is one of the most common measurements used in link prediction problem. Having a large number of the common neighbors easily create a link.
- Jaccard Coefficient (JC): It is a normalized form of the CN index.
- Adamic-Adar Coefficient (AA) [1]: It evaluates the impotency of a node when having less number of neighbors when predicting links.
- Preferential Attachment (PA) [3]: It generates the belief that nodes with large number of neighbors are more likely to form more in the future.
- node2vec [7]: It is a node embedding method, which learns nodes representation of network by preserving higher-order proximity between nodes. It used a higher probability of node occurrence in a fixed-length random walk.

- LINE [24]: It used an objective function to preserves the first-order and second-order neighborhoods to learn node representations, most similar to node2vec. It is useful to apply for large-scale network embedding.
- DeepWalk [20]: It used random walk model to learn vertex representations. This embedding can be used to predict link existence.
- SDNE [26]: It used both the first-order and second-order proximities together in an autoencoder based deep model to generate the vertex representations.

Regarding the implementations, we evaluated the Link prediction problem from the original code by the authors for node2vec[3], LINE[4], DeepWalk[5] and SDNE[6].

Dataset	**CN**	**JC**	**AA**	**PA**	**node2vec**	**LINE**	**DeepWalk**	**SDNE**	**LATB**
Enron	0.8106	0.8751	0.8970	0.8442	0.7596	0.5042	0.7190	**0.9437**	0.9302
Radoslaw	0.8417	0.8307	0.9028	0.8753	0.7417	0.6153	0.7342	0.8709	**0.9457**
Contact	0.8457	0.9141	0.9142	0.9027	0.8741	0.7360	0.8451	0.9376	**0.9906**
CollegeMessages	0.5742	0.5774	0.5843	0.5901	0.7049	0.4905	0.7506	0.7806	**0.9576**
EU-core	0.9227	0.9302	0.9341	0.7553	0.8602	0.6587	0.8201	0.9574	**0.9626**
Mathoverflow	0.7774	0.7692	0.7430	0.7783	0.7478	0.6587	0.7456	0.9574	**0.9968**

Table 2: Experimental Results based on AUC by comparing to Classic and Embedding methods.

4.4 Results

Our model achieves a significant improvement compared to other methods in various dynamic networks except the dataset Enron, which is better in SDNE. The standard network embedding methods, node2vec and LINE, perform the worst than other methods in terms of AUC. LATB performs significantly better in Contact and Eu-Core networks above 96% while the classic and embedding methods achieve below 92%. Moreover, LATB has an AUC higher than 93% among all tested dynamic networks. We can conclude that LATB can perform well in both very sparse (Enron) and dense (Radoslaw) networks. Tables 2 presents the results comparison of other methods with LATB.

5 Conclusions

In this paper, we propose a model for link prediction in dynamic networks by analyzing temporal behaviors of vertices, named LATB. To model the evolving pattern of each vertex, we propose a new scoring method, which can engage

[3] https://github.com/aditya-grover/node2vec
[4] https://github.com/tangjianpku/LINE
[5] https://github.com/phanein/deepwalk
[6] https://github.com/xiaohan2012/sdne-keras

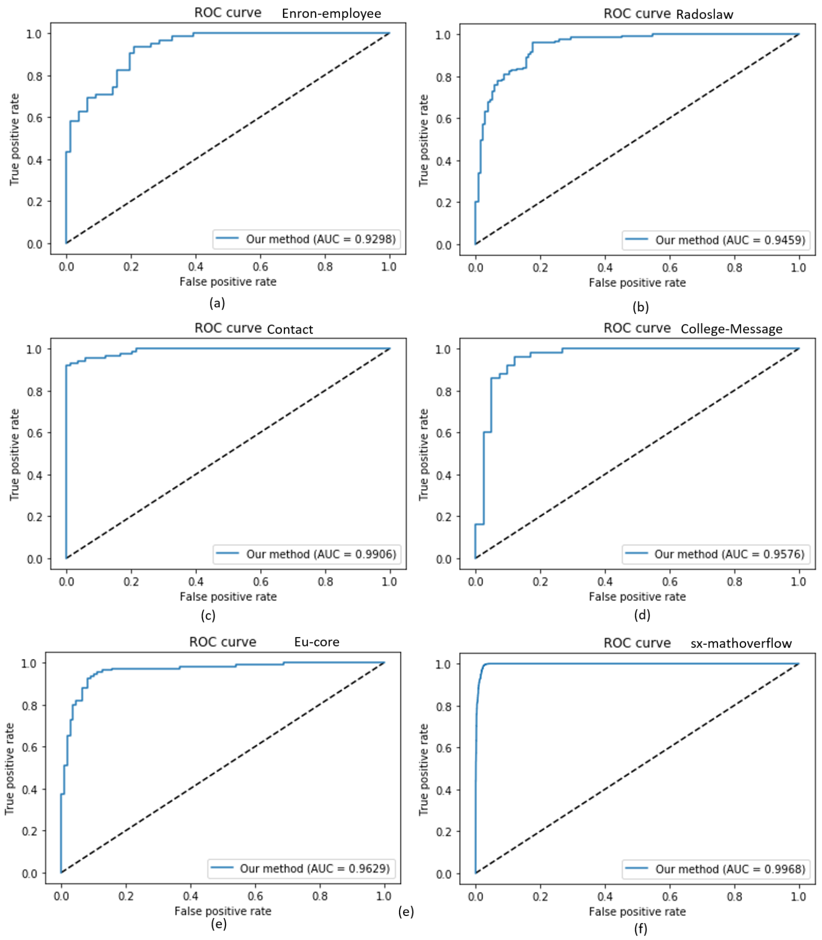

Fig. 3: Experiments on ROC curve. (a) ROC curve on College Message. (b) ROC curve on Enron-employee. (c) ROC curve on Radoslaw.(d) ROC curve on Contact. (e) ROC curve on Eu-Core.

the historical changes of vertices. To further address LATB, temporal changes of vertices are analyzed in two ways to measure the activeness of a node: how often a vertex interacts with existing connected nodes - to measure the strength of the relationship with its neighbors, and how fast and often collaborate with new nodes. Because both measures have a strong influence on deciding the node activeness, we introduce a probability function based on the activeness of nodes to evaluate the chances of being connected in the future. We also use two other weighted indexes: shortest distance and common neighbors, to incorporate the time-varying nature and number of link occurrences in neighbor nodes. In LATB, MLP is used as a classifier, which results in the status of link existence. Empirically, we compare LATB with classical methods and traditional embedding methods in five different real-world dynamic networks. Overall our model, LATB, achieves significant improvements and reaches above 93% of AUC.

Acknowledgment

This research work was supported by International Business Machines (IBM); experiments were conducted on a high performance IBM Power System S822LC Linux Server.

References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the web. Social networks **25**(3), 211–230 (2003)
2. Aydın, S.: Link prediction models for recommendation in academic collaboration network of Turkey. Ph.D. thesis (2017)
3. Barabâsi, A.L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A., Vicsek, T.: Evolution of the social network of scientific collaborations. Physica A: Statistical mechanics and its applications **311**(3-4), 590–614 (2002)
4. Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J.: Impact of human mobility on opportunistic forwarding algorithms. IEEE Transactions on Mobile Computing (6), 606–620 (2007)
5. Dorogovtsev, S.N., Mendes, J.F.: Evolution of networks. Advances in physics **51**(4), 1079–1187 (2002)
6. Goyal, P., Kamra, N., He, X., Liu, Y.: Dyngem: Deep embedding method for dynamic graphs. arXiv preprint arXiv:1805.11273 (2018)
7. 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. pp. 855–864. ACM (2016)
8. Huang, S., Tang, Y., Tang, F., Li, J.: Link prediction based on time-varied weight in co-authorship network. In: Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference on. pp. 706–709. IEEE (2014)
9. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1), 39–43 (1953)
10. Kaya, M., Jawed, M., Bütün, E., Alhajj, R.: Unsupervised link prediction based on time frames in weighted–directed citation networks. In: Trends in Social Network Analysis, pp. 189–205. Springer (2017)
11. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: European Conference on Machine Learning. pp. 217–226. Springer (2004)
12. Li, X., Du, N., Li, H., Li, K., Gao, J., Zhang, A.: A deep learning approach to link prediction in dynamic networks. In: Proceedings of the 2014 SIAM International Conference on Data Mining. pp. 289–297. SIAM (2014)
13. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American society for information science and technology **58**(7), 1019–1031 (2007)
14. Munasinghe, L., Ichise, R.: Time aware index for link prediction in social networks. In: International Conference on Data Warehousing and Knowledge Discovery. pp. 342–353. Springer (2011)

15. Murata, T., Moriyasu, S.: Link prediction of social networks based on weighted proximity measures. In: Proceedings of the IEEE/WIC/ACM international conference on web intelligence. pp. 85–88. IEEE Computer Society (2007)
16. Newman, M.E.: Clustering and preferential attachment in growing networks. Physical review E **64**(2), 025102 (2001)
17. Panzarasa, P., Opsahl, T., Carley, K.M.: Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. Journal of the American Society for Information Science and Technology **60**(5), 911–932 (2009)
18. Paranjape, A., Benson, A.R., Leskovec, J.: Motifs in temporal networks. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. pp. 601–610. ACM (2017)
19. Pavlov, M., Ichise, R.: Finding experts by link prediction in co-authorship networks. FEWS **290**, 42–55 (2007)
20. 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. pp. 701–710. ACM (2014)
21. Potgieter, A., April, K.A., Cooke, R.J., Osunmakinde, I.O.: Temporality in link prediction: Understanding social complexity. Emergence: Complexity & Organization (E: CO) **11**(1), 69–83 (2009)
22. Rosenblatt, F.: The perceptron: A theory of statistical separability in cognitive systems. United States Department of Commerce (1958)
23. Rossi, R., Ahmed, N.: The network data repository with interactive graph analytics and visualization. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
24. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web. pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)
25. Tylenda, T., Angelova, R., Bedathur, S.: Towards time-aware link prediction in evolving social networks. In: Proceedings of the 3rd workshop on social network mining and analysis. p. 9. ACM (2009)
26. 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. pp. 1225–1234. ACM (2016)
27. Wang, H., Shi, X., Yeung, D.Y.: Relational deep learning: A deep latent variable model for link prediction. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
28. Wang, P., Xu, B., Wu, Y., Zhou, X.: Link prediction in social networks: the state-of-the-art. Science China Information Sciences **58**(1), 1–38 (2015)
29. Wang, T., He, X.S., Zhou, M.Y., Fu, Z.Q.: Link prediction in evolving networks based on popularity of nodes. Scientific reports **7**(1), 7147 (2017)
30. Yang, S.H., Smola, A.J., Long, B., Zha, H., Chang, Y.: Friend or frenemy?: predicting signed ties in social networks. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. pp. 555–564. ACM (2012)
31. Yao, L., Wang, L., Pan, L., Yao, K.: Link prediction based on common-neighbors for dynamic social network. Procedia Computer Science **83**, 82–89 (2016)
32. Zhu, B., Xia, Y.: Link prediction in weighted networks: A weighted mutual information model. PloS one **11**(2), e0148265 (2016)