SCIENTIFIC REPORTS
natureresearch

OPEN

Identifying critical nodes in temporal networks by network embedding

En-Yu Yu[1], Yan Fu[1], Xiao Chen[2], Mei Xie[3] & Duan-Bing Chen[1,3,4]✉

Critical nodes in temporal networks play more significant role than other nodes on the structure and function of networks. The research on identifying critical nodes in temporal networks has attracted much attention since the real-world systems can be illustrated more accurately by temporal networks than static networks. Considering the topological information of networks, the algorithm *MLI* based on network embedding and machine learning are proposed in this paper. we convert the critical node identification problem in temporal networks into regression problem by the algorithm. The effectiveness of proposed methods is evaluated by SIR model and compared with well-known existing metrics such as temporal versions of betweenness, closeness, k-shell, degree deviation and dynamics-sensitive centralities in one synthetic and five real temporal networks. Experimental results show that the proposed method outperform these well-known methods in identifying critical nodes under spreading dynamic.

Complex networks are common in real life and can be used to represent complex systems in many fields[1]. Identifying critical nodes is an important topic in complex networks and it plays a crucial role in many applications, such as market advertising, rumor controlling and valuable scientific publication predicting[2,3]. In recent years, many methods are proposed to measure the importance of nodes in static networks. Among of these, degree centrality[4], semi-local centrality[5], k-shell[6] and H-index[7,8] are based on nodes' degrees; closeness[9], betweenness[10] and eccentricity centralities[11] are based on paths in networks; and PageRank[12], LeaderRank[13] and HITs[14] are based on eigenvector. However, the static networks whose edges are always active can not illustrate the dynamical systems. For this case, temporal networks are more suitable to present them[15–17]. Compared with static networks, temporal networks contain time information. The topology of temporal networks are changing with time and can describe many important activities in the real-word, including face-to-face conversations[18], the outbreak of epidemics[19] and the spread of news and ideas. Same as static networks, identifying critical nodes or influential spreaders on temporal networks also is a hot research topic. Up to now, there have been many methods focus on this problem. Kim et al. have presented a simple yet effective model, the time-ordered graph, which reduces a dynamic network to a static one with directed flows and defined the temporal version of degree, closeness and betweenness on temporal networks[20]. Taylor et al. proposed the eigenvector-based centrality for temporal networks[21], where the eigenvector and its components of a supra-centrality matrix can reflect the importance of nodes. Liu et al.[22] proposed dynamic-sensitive centrality to measure the importance of nodes in static networks and temporal dynamic-sensitive centrality[23] has advantage over the method proposed by Liu et al, which based on Markov chain for the epidemic model and derive the analytical result of node influence.

Although there are many methods to measure the influence of nodes in static or temporal networks, most of them can be regarded as a mission to find what kind of structure will make the node more influential. In this paper, we convert the critical node identification problem in temporal networks into regression problem by network embedding[24–26]. Network embedding assigns nodes in a network to low-dimensional representations and effectively preserves the network structure[27]. Recently, significant progresses have been made toward this emerging network analysis paradigm[28]. And network embedding has been used in community detection[29,30], link prediction[17] and etiology of diseases[31]. In terms of node embedding, Niepert et al. proposed a framework for

[1]Big Data Research Center, University of Electronic Science and Technology of China, Chengdu 611731, People's Republic of China. [2]Information Assurance Office of Army Staff, Beijing 100043, People's Republic of China. [3]The Center for Digitized Culture and Media, University of Electronic Science and Technology of China, Chengdu 611731, People's Republic of China. [4]Union Big Data Tech. Inc., Chengdu 610041, People's Republic of China. ✉email: dbchen@uestc.edu.cn

Network	N	E	L	δ
Training	1000	29940	30	*N/A*
TSF	500	14970	30	*N/A*
FRI	180	45070	30	2 hours
Contact	274	28244	30	10 minutes
Hypertext	410	20818	30	12 hours
DNC	2029	39264	30	16 days
UCS	1899	59835	30	6 days

Table 1. Statistical properties of the networks used in the experimental analysis. N and E are the total number of nodes and edges, respectively. L is the number of time points and δ is the time interval.

learning convolutional neural networks for arbitrary graphs[32], presenting a general approach to extract locally connected regions from networks. *node2vec*[33] is an algorithmic framework for learning a low-dimensional representations of nodes in complex networks and the mapping of nodes to a low-dimensional space of features can be learned by maximizes the likelihood of preserving network neighborhoods of nodes. Kipf et al. presented a scalable approach for semi-supervised learning on complex networks which is based on an efficient variant of convolutional neural networks[34]. In order to explore the impact of temporal information on the importance of the nodes, Qu et al.[35] proposed a temporal information gathering (TIG) process for evaluating the significance of the nodes in temporal networks. The key to the TIG process is that the importance of a node depends on the importance of its neighborhood. Qi et al.[36] presented a Deep Autoencoding Gaussian Mixture Model (DAGMM) for unsupervised anomaly detection. In DAGMM, low-dimensional representations of nodes are generated by a deep autoencoder and applied to rank nodes in temporal networks.

By using the topological information, we proposed the algorithm based on network embedding and machine learning. The performance of proposed method is compared with that of temporal versions of betweenness centrality, closeness centrality[37], k-shell[38], degree deviation centrality[39] and dynamics-sensitive centrality[23] in SIR model[40,41] for one synthetic and five real temporal networks. The results show that the proposed method in this paper can effectively identify critical nodes which have greater impacts on information spreading in temporal networks.

Results

The real spreading ability of a node is estimated by SIR spreading model in this paper. The performance of *MLI* is evaluated by SIR spreading model, and compared with well-known existing metrics such as temporal versions of betweenness centrality, closeness centrality, k-shell, degree deviation centrality and dynamics-sensitive centrality in one synthetic and five real temporal networks. In all experiments, we set $D = 8, \alpha = 0.2$.

Data sets.

Training Data Set. A temporal scale-free network generated by BA model[42]. In addition, using other network generation models can also get good training results. We use the BA model in this paper, and other models can be used, such as WS models[43], etc.

Testing Data Sets. Six temporal networks are used to evaluate the performance of the methods.

(1) Temporal scale-free network(TSF). This undirected network is a combination of 30 snapshots, and each snapshot is generated by BA model[42].
(2) High school friendship relations network(FRI). This undirected network is a medium-sized data set correspond to the contacts and friendship relations between students in a high school[44].
(3) Contact network(Contact). An undirected network represents contacts between users measured by carried wireless devices[45].
(4) Sociopatterns-hypertext network(Hypertext). This is an undirected network of face-to-face contacts of the attendees of the ACM Hypertext 2009 conference[46].
(5) DNC co-recipient network(DNC). This is the undirected network of emails in the 2016 Democratic National Committee email leak[47].
(6) UC Irvine messages network(UCS). An undirected network contains sent messages between the users of an online community of students from the University of California, Irvine[48].

In Table 1, some detailed statistical properties of above networks are listed. In this paper, these networks are both undirected and unweighted. FRI, Contact, Hypertext, DNC, UCS are real-world networks which represent human interactions in diverse social systems and have different topological and temporal characteristics, Training and TSF are generated by BA model. All temporal networks are divided into 30 snapshots.

Spreading performance.

In order to evaluate the performance of *MLI* on different β and β_t. The kendall's tau coefficients between the ranking scores and the real infected scales on different β and β_t is shown in fig. 1.

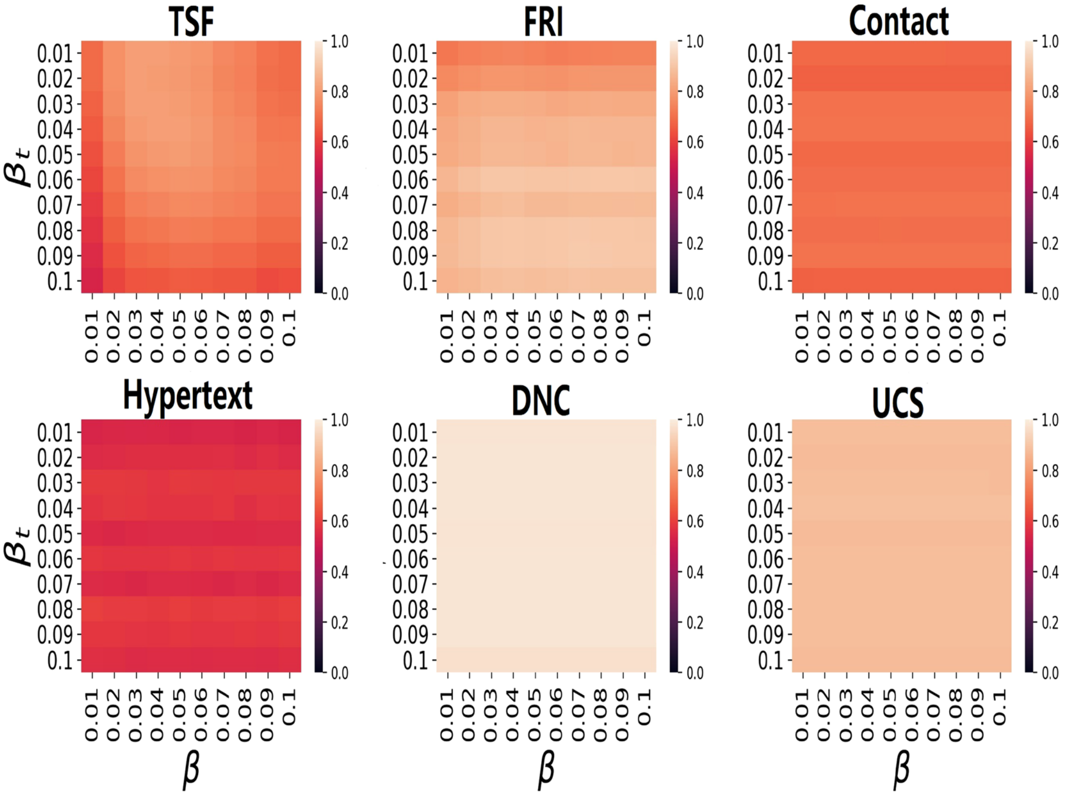

Figure 1. The heat map of kendall's tau correlation coefficients between the ranking scores and real infected scales with varying infection rate β and train infection rate β_t by *MLI*.

From fig. 1, it can be seen that there is a high value in any position of these heat maps. When infection rate β and train infection rate β_t varies from 0.01 to 0.10, the performance of *MLI* will vary slightly with the change of β_t.

As mentioned in above analysis, *MLI* performs well in most cases with slight influence of train infection rate β_t. So we can fix train infection rate β_t as 0.1 in the following analysis. In order to evaluate the performance of these algorithms under different infection rates, *MLI* is compared with other five methods. From fig. 2, it can be seen that *MLI* has the maximum value under the infection rate varying from 0.01 to 0.1. It means that the ranking result of *MLI* is the closest to that of the real infected scales in all methods.

We also use the top k comparison method in this paper since we often care about top-ranked nodes. For details, sort nodes in descending order according to their ranking score firstly, and compare the top k nodes obtained by methods with real top k nodes simulated by SIR spreading model. The evaluation index is hitting rates(*HR*), which is defined as

$$HR = \frac{|C \bigcap R|}{|R|}, \tag{1}$$

where C and R are top k nodes obtained by algorithms and SIR spreading model respectively and $|\cdot|$ is the size of set. The higher the *HR* is, the better the performance of the algorithm is. From fig. 3, it can be seen that *MLI* has the maximum *HR* in most cases when find the top 10% critical nodes under the infection rate varying from 0.01 to 0.1. These results demonstrate that *MLI* can find critical nodes under different infection rates.

Discussion

How to identifying critical nodes in temporal networks is an interesting and important topic in many applications. Most of previous researches concentrate on finding what kind of structure will make the node more influential. Inspired by the concept of network embedding and machine learning, the algorithm *MLI* is proposed in this paper. According to the experimental results on five real-world and a synthetic temporal networks, *MLI* performs much better than other five benchmark methods in identifying the nodes which have great impact on information spreading. We can use *MLI* to detect the potential super-spreaders for epidemic control in temporal networks. *MLI* enable us to investigate the dynamics in spreading process and the results of parameters analysis show that *MLI* outperform other five methods significantly in most cases with fixing train infection rate β_t. What's more, *MLI* has a low computational complexity and can be used in large-scale networks. The method presented in this paper have provided a new idea for identifying critical nodes in complex networks, and the method can be extended to many other dynamical analyses such as the impact of edges on spreading dynamics.

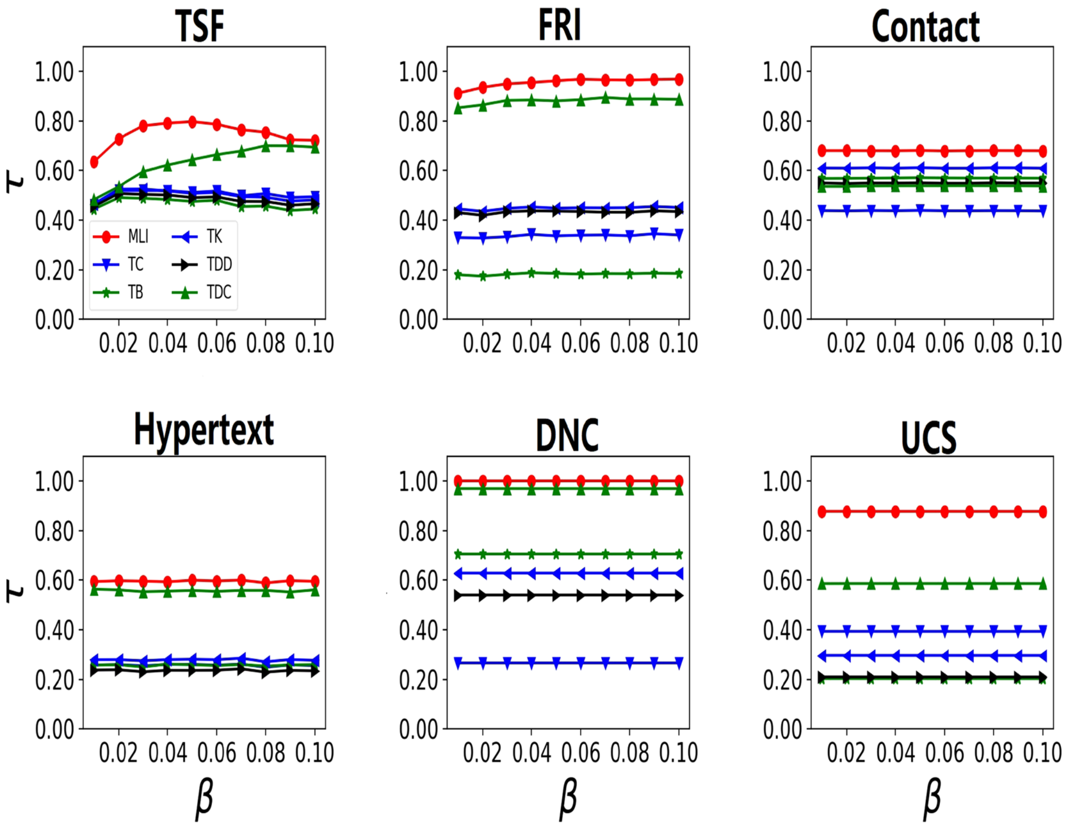

Figure 2. kendall's tau correlation coefficients between the ranking scores and the relative differences of real infected scales with varying infection rate β.

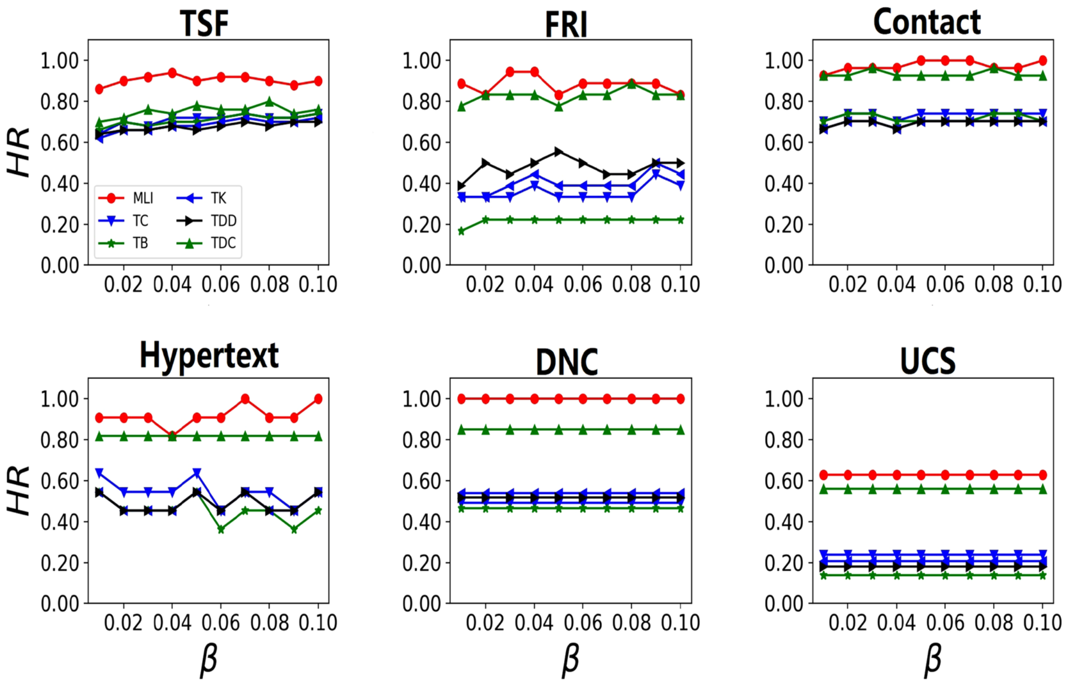

Figure 3. The top10% *HR* under different infection rate β.

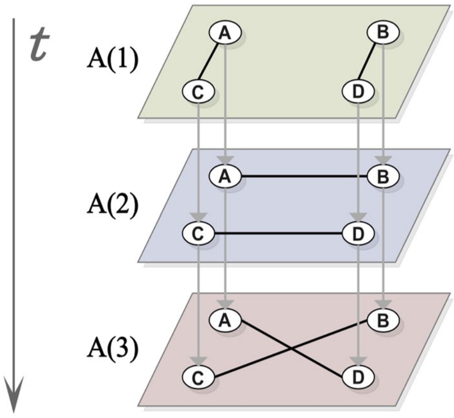

Figure 4. An example temporal network represented by using a multilayer network structure.

Methods

Temporal networks.

A temporal network with N nodes and time length T can be defined as a sequence of L snapshots with $\delta = T/L$. That is, a temporal network can be represented as a series of static graphs $G_1, G_1, \ldots, G_L$. Where $G_t (1 \leqslant t \leqslant L)$ represents the aggregate graph which consists of N nodes and a set of edges E_t where an edge $e(u, v) \in E_t$ only if node u has a connection with node v at snapshot t (a time interval $[\delta(t-1), \delta t]$). In this study, all edges are undirected. Fig. 4 is an example temporal network represented by a multilayer network structure. There are four nodes A, B, C, D with $T = 3$, $L = 3$ and $\delta = 1$.

Convolutional Neural Networks.

CNNs are kind of Feedforward neural networks with convolutional computation and deep structure. It is one of the representative algorithms of deep learning. CNN is an incredibly successful technology that has been applied to computer vision and natural language processing[49,50]. CNNs in essence are neural networks which use the convolution operation as one of their layers. A typical CNN consists of several convolution and pooling layers. The purpose of the first convolutional layer is to extract a common pattern found within localized regions of the input data. CNNs convolve learned filters over the input data, computing the inner product and outputting the result as tensors whose depth is the number of filters.

Benchmark methods.

We consider some benchmark methods in this paper, Tang et al.[37] proposed a method to identify important nodes using temporal versions of conventional centrality metrics, Kim et al.[20] extend Tang's work to a more general and more realistic model. Readers can refer to Tang et al.[37] or Kim et al.[20] for details of the temporal versions of conventional centrality. In this paper, we consider *TC* and *TB* proposed by Kim et al.[20], *TK* proposed by Wang[39], *TDD* proposed by Ye[38] and *TDC* proposed by Huang[23]. For all methods, we calculate the value in a time interval [0, *T*] with *L* time snapshots.

The temporal version of closeness centrality(*TC*) is defined as

$$TC(v) = \sum_{0 \leqslant t < T} \sum_{u \in V \backslash \{v\}}^{\frac{1}{\Delta_{t,T}(v,u)}}, \tag{2}$$

where $\Delta_{t,T}(v, u)$ is the distance of temporal shortest path from node v to node u on a time interval $[t, T]$. If there is no temporal path from v to u on a time interval $[t, T]$, $\Delta_{t,T}(v, u)$ is infinity.

The temporal version of betweenness centrality(*TB*) is defined as

$$TB(v) = \sum_{0 \leqslant t < T} \sum_{\substack{s \neq v \neq d \in V \\ \sigma_{t,T}(s,d) > 0}} \frac{\sigma_{t,T}(s, d, v)}{\sigma_{t,T}(s, d)}, \tag{3}$$

where $\sigma_{t,T}(s, d)$ denotes the number of temporal shortest paths from source s to destination d on the time interval $[t, T]$, $\sigma_{t,T}(s, d, v)$ is the number of paths from source s to destination d on the time interval $[t, T]$ with v in their interior.

The temporal k-shell(*TK*) is defined as

$$TK(v) = \sum_{u \in \Gamma_v} \sum_{t=1}^{L} min\{k_s^t(v), k_s^t(u)\}, \tag{4}$$

where $k_s^t(v)$ denote the k-shell score of node v in the time snapshot t, Γ_v is node v's neighbors in the slice network.

The temporal degree deviation centrality(*TDD*) is defined as

$$TDD(v) = \sqrt{\frac{1}{L}\sum_{t=1}^{L}(D_t(v) - D(v))^2}, \tag{5}$$

where $D_t(v)$ denote the degree of node v in the time snapshot t, $D(v)$ is the average number of $D_t(v)$ in all slice networks.

The temporal dynamics-sensitive centrality(TDC) of all nodes can be described by the vector

$$S = \sum_{r=0}^{L-1} \beta H_*^r A(r+1)V \quad , \tag{6}$$

$$H_*^t = \prod_{\alpha=t}^{1}[\beta A(\alpha) + (1-\mu)I], H_*^0 = 1 \quad , \tag{7}$$

where S is a vector and denote the spreading influence of all nodes, $A(t)$ is the adjacency matrix in time snapshot $t, V = (1, 1, \ldots, 1)^T$, β and μ are infection probability and recover probability in SIR spreading model.

SIR spreading model. In SIR model[51], there are three states: (1) $S(t)$ denotes the number of nodes which may be infected(not yet infected); (2)$I(t)$ denotes the number of nodes which have been infected and will spread the disease or information to susceptible nodes; (3)$R(t)$ denotes the number of nodes which have been recovered from the disease or boredom the information and will never be infected by infected nodes again. In a network, each infected node will infect all susceptible neighbors with a certain probability β. Infected nodes will be recovered with probability μ(for simplicity, $\mu = 1$ in this paper)at each step. The process is repeated within the given time step $t(t \leqslant$L). $N_v(t)$ is defined as the number of infected nodes after t steps under the disease spreads from the initial node v firstly. We can use $N_v(L)$ to represent the finally infected scale of node v in this paper.

MLI model. With the development of machine learning, some researchers have proposed the concept of network embedding, which aims at learning low-dimensional latent representation of nodes in a network. These representations can be used as features for a wide range of tasks on graphs such as classification, clustering, link prediction and visualization. Inspired by this concept, we combine network embedding and machine learning to identify critical nodes in temporal networks. For details, the method *MLI*(machine learning index) can be described by following steps:

(1) Feature Matrices and Labels: Similar to convolutional neural network for images, we need to input feature matrices and labels to training model. For each node, we construct a feature matrix by its neighborhoods in all snapshots. And obtaining all nodes' infected scale by SIR spreading model(training infection rate is β_t and training recovery rate is $\mu_t = 1$) as the labels. For details, the node embedding algorithm is shown in Algorithm. 1. When finding neighbors, nearer neighbors have higher priority to choose, that is, 1-hop neighbors have priority over 2-hop neighbors, if two nodes have equal hops, the node with larger degree has been chosen preferential.

(2) Convolutional Architecture: The convolutional neural network in this paper have 2 convolutional layers, 2 pooling layers and 1 fully-connected layer. In the first convolutional layer, kernel size is 5×5, input channel is 1 and output channels are 16, stride is 1 and paddings are 2. In the second convolutional layer, kernel size is 5×5, input channels are 16 and output channels are 32, stride is 1 and paddings are 2. 2 pooling layers are 2×2 max pooling and the fully-connected layer is $32 * (D/4) * (D/4) \times 1$. The activation function is ReLU and the loss function is squared loss function.

Algorithm 1 Generate feature matrices for nodes.

```
Input: Temporal network $TG = [G_1, G_1, ..., G_L]$ with the set of nodes $V = \{v_1, ..., v_n\}$ and the set of edges $E \subseteq V \times V$;
    Parameters $D, \alpha$;
Output: Feature matrices for all nodes $FM$;
for each node $v$ in $V$ do
    $FM[v] = \emptyset$
end for
for each node $v$ in $V$ do
    for each graph $G$ in $TG$ do
        find $D-1$ neighbors $v_1, ..., v_{D-1}$ for $v$, let $v_0, v_1, ..., v_{D-1}$ be node $u$ and its $D-1$ neighbors;
        generate subnetwork $G_v$ contains node $v$ and its $D-1$ neighbors, $\mathbf{A}_v(D \times D)$ is the adjacency matrix of $G_v$;
        $\mathbf{B}_v = \mathbf{A}_v, \mathbf{B}_v[i,i] = k_{v_i}, i \in [0, D-1]$, $k_{v_i}$ is degree of node $v_i$ in $G$;
        $FM[v] = FM[v] + \alpha \mathbf{B}_v$
    end for
end for
    return $FM$;
```

(3) Ranking Nodes: Firstly, use training sets (generate feature matrices and labels for all nodes) to train the CNN model. Secondly, select arbitrary networks, generate feature matrices for all nodes without labels and get the score of these nodes by the trained CNN model. Finally, ranking the nodes by its score.

Time complexity. We compare the time complexity of the above six methods. Although *MLI* needs time to generate training set and train the parameters, we can use the parameters for all temporal networks. Let the size of feature matrices be $D \times D(D << N)$, the temporal network has N nodes and L snapshots. The time complexity of generating feature matrices is $O(NL)$,the time complexity of training[52] is $O(I \cdot \sum_{l=1}^{P} M_l^2 \cdot K_l^2 \cdot C_{l-1} \cdot C_l)$, where I is the number of iterations. P is the number of convolutional layers. M_l is the side length of the output feature maps of convolutional kernels at the l^{th} convolutional layer. K_l and C_l are the side length of convolutional kernels and the number of output channels at the l^{th} convolutional layer, respectively. So the time complexity of *MLI* in new temporal networks is $O(NL + \sum_{l=1}^{P} M_l^2 \cdot K_l^2 \cdot C_{l-1} \cdot C_l)$.

In addition, the time complexity of *TDD* is $O(m)$. We need take $O(m)$ to calculate the k-shell score of all nodes in all snapshots. So the time complexity of *TK* is also $O(m)$. And the time complexity of *TDC* is $O(n^2)$ because we need take $O(n^2)$ to calculate the multiplication of sparse matrices. The time complexity of *TC* is $O(mn^2)$ and the time complexity of *TB* is $O(m^3n^3)$[20]. From above analysis, it can be seen that MLI_{LM} has a low computational complexity and can be used in large-scale networks.

Received: 9 October 2019; Accepted: 22 June 2020
Published online: 27 July 2020

References

1. Newman, M. E. The structure and function of complex networks. *SIAM Rev.* **45**, 167–256 (2003).
2. Zhang, T. *et al.* A discount strategy in word-of-mouth marketing. *Communications in Nonlinear Science and Numerical Simulation* (2019).
3. Zhou, F., Lü, L. & Mariani, M. S. Fast influencers in complex networks. *Communications in Nonlinear Science and Numerical Simulation* (2019).
4. Bonacich, P. Factoring and weighting approaches to status scores and clique identification. *J. Math. Soc.* **2**, 113–120 (1972).
5. Chen, D., Lü, L., Shang, M.-S., Zhang, Y.-C. & Zhou, T. Identifying influential nodes in complex networks. *Phys. Stat. Mech. Appl.* **391**, 1777–1787 (2012).
6. Kitsak, M. *et al.* Identification of influential spreaders in complex networks. *Nat. Phys.* **6**, 888 (2010).
7. Lü, L., Zhou, T., Zhang, Q.-M. & Stanley, H. E. The h-index of a network node and its relation to degree and coreness. *Nat. Commun.* **7**, 10168 (2016).
8. Pastor-Satorras, R. & Castellano, C. Topological structure and the h index in complex networks. *Phys. Rev. E* **95**, 022301 (2017).
9. Freeman, L. C. Centrality in social networks conceptual clarification. *Soc. Netw.* **1**, 215–239 (1978).
10. Freeman, L. C. A set of measures of centrality based on betweenness. *Sociometry* **35–41**, (1977).
11. Hage, P. & Harary, F. Eccentricity and centrality in networks. *Soc. Netw.* **17**, 57–63 (1995).
12. Brin, S. & Page, L. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.* **30**, 107–117 (1998).
13. Lü, L., Zhang, Y.-C., Yeung, C. H. & Zhou, T. Leaders in social networks, the delicious case. *PLoS One* **6**, e21202 (2011).
14. Kleinberg, J. M. Authoritative sources in a hyperlinked environment. *J. ACM (JACM)* **46**, 604–632 (1999).
15. Holme, P. & Saramäki, J. Temporal networks. *Phys. Rep.* **519**, 97–125 (2012).
16. Li, M., Rao, V. D., Gernat, T. & Dankowicz, H. Lifetime-preserving reference models for characterizing spreading dynamics on temporal networks. *Sci. Rep.* **8**, 709 (2018).
17. Ma, X., Sun, P. & Wang, Y. Graph regularized nonnegative matrix factorization for temporal link prediction in dynamic networks. *Phys. A Stat. Mech. Appl.* **496**, 121–136 (2018).
18. Takaguchi, T., Nakamura, M., Sato, N., Yano, K. & Masuda, N. Predictability of conversation partners. *Phys. Rev. X* **1**, 011008 (2011).
19. Lloyd-Smith, J. O. *et al.* Epidemic dynamics at the human–animal interface. *Science* **326**, 1362–1367 (2009).
20. Kim, H. & Anderson, R. Temporal node centrality in complex networks. *Phys. Rev. E* **85**, 026107 (2012).
21. Taylor, D., Myers, S. A., Clauset, A., Porter, M. A. & Mucha, P. J. Eigenvector-based centrality measures for temporal networks. *Multiscale Model. Simul.* **15**, 537–574 (2017).
22. Liu, J.-G., Lin, J.-H., Guo, Q. & Zhou, T. Locating influential nodes via dynamics-sensitive centrality. *Sci. Rep.* **6**, 21380 (2016).
23. Huang, D.-W. & Yu, Z.-G. Dynamic-sensitive centrality of nodes in temporal networks. *Sci. Rep.* **7**, 41454 (2017).
24. Cui, P., Wang, X., Pei, J. & Zhu, W. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering* (2018).
25. Wang, Y., Yao, Y., Tong, H., Xu, F. & Lu, J. A brief review of network embedding. *Big Data Min. Anal.* **2**, 35–47 (2018).
26. Zhang, Z., Cui, P. & Zhu, W. Deep learning on graphs: A survey. arXiv preprint arXiv:1812.04202 (2018).
27. Lai, K.-H., Chen, C.-M., Tsai, M.-F. & Wang, C.-J. Navwalker: Information augmented network embedding. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 9–16 (IEEE, 2018).
28. Zhou, J. *et al.* Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434 (2018).
29. Ma, X., Dong, D. & Wang, Q. Community detection in multi-layer networks using joint nonnegative matrix factorization. *IEEE Transactions on Knowledge & Data Engineering* **1–1**, (2018).
30. Ma, X. & Dong, D. Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks. *IEEE Trans. Knowl. Data Eng.* **29**, 1045–1058 (2017).
31. Ma, X., Tang, W., Wang, P., Guo, X. & Gao, L. Extracting stage-specific and dynamic modules through analyzing multiple networks associated with cancer progression. *IEEE/ACM Transactions on Computational Biology & Bioinformatics* **1–1**, (2016).
32. Niepert, M., Ahmed, M. & Kutzkov, K. Learning convolutional neural networks for graphs. *International conference on machine learning* **2014–2023**, (2016).
33. 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*, 855–864 (ACM, 2016).
34. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016).
35. Qu, C., Zhan, X., Wang, G., Wu, J. & Zhang, Z.-K. Temporal information gathering process for node ranking in time-varying networks. *Chaos Interdiscip. J. Nonlinear Sci.* **29**, 033116 (2019).
36. Qi, S. *et al.* Tgnet: Learning to rank nodes in temporal graphs. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 97–106 (ACM, 2018).

37. Tang, J., Musolesi, M., Mascolo, C., Latora, V. & Nicosia, V. Analysing information flows and key mediators through temporal centrality metrics. In *Proceedings of the 3rd Workshop on Social Network Systems*, 3 (ACM, 2010).
38. Ye, Z., Zhan, X., Zhou, Y., Liu, C. & Zhang, Z.-K. Identifying vital nodes on temporal networks: An edge-based k-shell decomposition. In *2017 36th Chinese Control Conference (CCC)*, 1402–1407 (IEEE, 2017).
39. Wang, Z., Pei, X., Wang, Y. & Yao, Y. Ranking the key nodes with temporal degree deviation centrality on complex networks. In *2017 29th Chinese Control And Decision Conference (CCDC)*, 1484–1489 (IEEE, 2017).
40. Kimura, M., Saito, K. & Motoda, H. Blocking links to minimize contamination spread in a social network. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **3**, 9 (2009).
41. Newman, M. E. Spread of epidemic disease on networks. *Phys. Rev. E* **66**, 016128 (2002).
42. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509–512 (1999).
43. Watts, D. J. & Strogatz, S. H. Collective dynamics of small world networks. *Nature* **393**, 440–442 (1998).
44. Mastrandrea, R., Fournet, J. & Barrat, A. Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PLoS One* **10**, e0136497 (2015).
45. Chaintreau, A. *et al.* Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing* **606–620**, (2007).
46. Isella, L. *et al.* Whats in a crowd? Analysis of face-to-face behavioral networks. *J. Theoret. Biol.* **271**, 166–180 (2011).
47. Dnc co-recipient network dataset – KONECT (2016).
48. Opsahl, T. & Panzarasa, P. Clustering in weighted networks. *Soc. Netw.* **31**, 155–163 (2009).
49. Atlas, L. E., Homma, T. & Marks, R. J. II. An artificial neural network for spatio-temporal bipolar patterns: Application to phoneme classification. *Neural Information Processing Systems* **31–40**, (1988).
50. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
51. Yu, E.-Y., Chen, D.-B. & Zhao, J.-Y. Identifying critical edges in complex networks. *Sci. Rep.* **8**, 1–8 (2018).
52. He, K. & Sun, J. Convolutional neural networks at constrained time cost. *Proceedings of the IEEE conference on computer vision and pattern recognition* **5353–5360**, (2015).

Acknowledgements

This work is partially supported by the National Natural Science Foundation of China under Grant Nos. 61673085 and 61433014, by Science Strength Promotion Programme of UESTC under Grant No. Y03111023901014006, and by the Fundamental Research for the Central Universities under Grant No. ZYGX2019J074.

Author contributions

E.-Y. Y. , D.-B. C. , Y. F. , X. C. and M. X. designed the method and prepared all figures. E.-Y. Y. performed the experiments and analyzed the data. All authors wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to D.-B.C.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.