PLOS ONE

RESEARCH ARTICLE

Diffusion characteristics classification framework for identification of diffusion source in complex networks

Fan Yang[1,2], **Jingxian Liu**[1,2], **Ruisheng Zhang**[3], **Yabing Yao**[4]*

1 Key Laboratory of Intelligent Information Processing and Graph Processing, Guangxi University of Science and Technology, Liuzhou, Guangxi, China, **2** School of Computer Science and Technology, Guangxi University of Science and Technology, Liuzhou, Guangxi, China, **3** School of Information Science and Engineering, Lanzhou University, Lanzhou, Gansu, China, **4** School of Computer and Communication, Lanzhou University of Technology, Lanzhou, Gansu, China

* yaoyabing@lut.edu.cn

Citation: Yang F, Liu J, Zhang R, Yao Y (2023) Diffusion characteristics classification framework for identification of diffusion source in complex networks. PLoS ONE 18(5): e0285563. https://doi.org/10.1371/journal.pone.0285563

Editor: Vincenzo Bonnici, University of Parma, ITALY

Received: November 20, 2022

Accepted: April 26, 2023

Published: May 15, 2023

Data Availability Statement: All real dataset in the experiments are available from the networkrepository database. 1. https://networkrepository.com/ca-netscience.php 2. https://networkrepository.com/subelj-euroroad.php 3. https://networkrepository.com/email-univ.php 4. https://doi.org/10.1007/978-3-642-01206-8_5 Source: 1. Rossi RA, Ahmed NK. The Network Data Repository with Interactive Graph Analytics and Visualization. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. Austin, Texas; 2015. p. 4292|-4293. Available from: https://ojs.aaai.org/index.php/AAAI/article/view/9277. 2.

Abstract

The diffusion phenomena taking place in complex networks are usually modelled as diffusion process, such as the diffusion of diseases, rumors and viruses. Identification of diffusion source is crucial for developing strategies to control these harmful diffusion processes. At present, accurately identifying the diffusion source is still an opening challenge. In this paper, we define a kind of diffusion characteristics that is composed of the diffusion direction and time information of observers, and propose a neural networks based diffusion characteristics classification framework (NN-DCCF) to identify the source. The NN-DCCF contains three stages. First, the diffusion characteristics are utilized to construct network snapshot feature. Then, a graph LSTM auto-encoder is proposed to convert the network snapshot feature into low-dimension representation vectors. Further, a source classification neural network is proposed to identify the diffusion source by classifying the representation vectors. With NN-DCCF, the identification of diffusion source is converted into a classification problem. Experiments are performed on a series of synthetic and real networks. The results show that the NN-DCCF is feasible and effective in accurately identifying the diffusion source.

Introduction

Most complex systems in the real world take the form of networks [1] in which the nodes and edges denote the units and the interactions between units, respectively. Various diffusion phenomena taking place in networks are usually modelled as diffusion process [1], such as disease spreading [2], rumor diffusion [3] and computer virus propagation [4]. The ubiquity of these harmful diffusion processes has incurred huge losses to human society. Therefore, it is of great theoretical and practical significance to develop effective strategies to control the harmful diffusion process. One of the important measures is identifying the diffusion source that initiates the diffusion process on networks, which has attracted widespread attentions in recent years

Gregory S. Finding overlapping communities using disjoint community detection algorithms. In: Complex networks; 2009. p. 47-61. The synthetic dataset in the experimants can be generated by igraph: https://igraph.org/ The related theory can be found in: 1. Barabasi AL, Albert R. Emergence of Scaling in Random Networks. Science.1999;286 (5439):509-512. 2. Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. Nature.1998;393(6684):440. doi:https://doi.org/10.1038/30918.

Funding: This work was supported by: 1. National Natural Science Foundation of China (Grant No. 62062010) (Fan Yang). https://www.nsfc.gov.cn/ 2. Science and Technology Planning Project of Guangxi (Grant No. AD19245101) (Fan Yang). http://kjt.gxzf.gov.cn/ 3. Science and Technology Planning Project of Liuzhou City (Grant No. 2020PAAA0606) (Fan Yang). http://kjj.liuzhou.gov.cn/ 4. Higher Education Innovation Fund project of Gansu (No. 2022A-022) (Yabing Yao). http://jyt.gansu.gov.cn/ 5. National Natural Science Foundation of China (Grant No. 62061003) (Jingxian Liu). https://www.nsfc.gov.cn/ 6. Doctoral Foundation of Guangxi University of Science and Technology (Grant No. 19Z06) (Fan Yang). https://www.gxust.edu.cn/ 7. Longyuan Youth Innovation and Entrepreneurship Talents Team Project of Gansu (No. 2021LQTD24) (Yabing Yao). http://www.gszg.gov.cn/ The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

[5]. Many existing source identification methods provided effective solutions for some important issues in reality, such as identifying the source of SARS [6], COVID-19 [7], Cholera [8], finding the source of foodborne disease [9], etc. However, accurately identifying the diffusion source is still an opening challenge.

The success of artificial neural networks has boosted research on many scientific fields [10–12]. Especially, the emergence of graph neural networks [13, 14] (GNNs) and network embedding [15, 16] facilitate the applications of artificial neural networks on irregular structures of networks. GNNs are the neural network models to address different graph tasks in an end-to-end way [13]. The most common GNNs include recurrent graphs neural networks [17], convolutional graph neural networks [18], graph autoencoders [13], etc. Network embedding is composed of various kinds of methods designed for a same task, i.e., network representation learning [13]. Recently, GNNs and network embedding have been successfully introduced into some important issues of complex networks [13, 16], such as link prediction and node classification. However, only a few artificial neural networks based methods focused on the diffusion source identification problem [19, 20]. Li et al. [19] proposed a label propagation framework to locate the diffusion source. Due to the common characteristics between label propagation framework and graph convolutional networks (GCNs), the source identification is converted into a multi-classification problem. Dong et al. [20] detected multiple sources by utilizing the wavefront information. Since existing GNNs is not a suitable solution for the wavefront based method, they developed a novel multi-task learning model based on encoder-decoder structure. Different from the two methods in [19] and [20], this paper utilizes the diffusion time and direction information recorded in limited observers to identify the diffusion source. The two types of information have been proved to be helpful in accurately identifying the source [21–26]. We define the two types of information as diffusion characteristics, and identify the diffusion source by classifying the diffusion characteristics. Although existing GNNs and network embedding are powerful models to process graph data, both of them are not suitable to be used to process the diffusion characteristics which is dynamically generated in a diffusion process. Therefore, we develop a novel neural networks based diffusion characteristics classification framework, which contains the following three stages, (*i*) the diffusion characteristics are utilized to construct network snapshot feature, (*ii*) a graph LSTM auto-encoder is proposed, by which the network snapshot feature is represented as low-dimension vectors, (*iii*) a source classification neural network is proposed to identify the diffusion source by classifying the representation vectors of network snapshot feature. With the proposed framework, the identification of diffusion source is converted into a classification problem. Further, the feasibility and effectiveness of this framework is validated by the experimental results.

The rest of this paper is organized as follows. Existing related works are briefly reviewed in Section Related work. The neural networks based diffusion characteristics classification framework is proposed in Section Materials and methods. The experimental results are discussed in Section Results. We conclude this work in Section Conclusion.

Related work

The early diffusion source identification methods were developed for unweighted networks. A systematic method was pioneered by Shah et al. [27], they constructed a source estimator based on a topological quantity termed as Rumor Centrality (RC). The RC has been extended to identify the diffusion source in more complex environments [28–30]. Zhu et al. [31] proposed a sample path based method termed as Jordan Center (JC), which has been improved to identify the diffusion source with limited observations [32–34]. Meanwhile, many methods based on various ideas were proposed for unweighted networks, including the Dynamic

Message Passing based method [35], the Belief Propagation based method [36], the Monte-Carlo method based method [37], the Rationality Observation based method [38], the Label Ranking framework based method [39], the Time Aggregated Graph based method [40], etc. The above methods are effective in unweighted networks. However, in reality, we have to consider various significant weights associated with the edges in networks, such as the traffic, the time delay and so on.

For weighted networks, Brockmann et al. [6] modeled the Global Mobility Network as a weighted graph, and identified the epidemic source based on a novel effective distance. This method has been extended to identify multi-source by Jiang et al. [41]. Meanwhile, several methods based on various ideas were proposed to identify the diffusion source in weighted networks [42–44]. However, these methods require the knowledge of all nodes state. In reality, it is often the case that only limited nodes state can be observed [45]. For this problem, many methods were proposed by utilizing limited observers, including the Time-Reversal Backward Spreading algorithm [24], the Backward Diffusion-based method [46], the improved Gaussian estimator [47], the Gromov matrix based method [25], the Greedy Optimization based algorithm [26], the Sequential Neighbour Filtering algorithm [48], the Estimated Propagation Delay based algorithms [49], etc. These methods [24–26, 46–49] mainly utilized the diffusion time information of observers to identify the source. Pinto et al. [50] proposed a Gaussian estimator, which is the first method to identify the source by utilizing the diffusion direction information of observers. However, the diffusion direction information is only used in the tree graphs. Yang et al. [21] improved the accuracy of Gaussian estimator on general graphs by utilizing the diffusion direction information of observers. Zhu et al. [22, 23] also proposed a path-based source identification method by utilizing the diffusion direction information of observers. Obviously, the diffusion time and direction information of observers play important roles in accurately identifying the diffusion source.

Different from all the traditional source identification methods mentioned above, in recent years, a few artificial neural networks based methods are developed to identify the source. Li et al. [19] proposed a Source Identification Graph Convolutional Network (SIGN) framework, this method requires the knowledge of complete observation. Dong et al. [20] proposed a graph constraint based sequential source identification model. To obtain the wavefront information, this method [20] also requires the knowledge of complete observation. However, in reality, it is often the case that only limited nodes state can be observed [45]. In this paper, we identify the diffusion source by utilizing limited observers. We define the diffusion time and direction information of observers as diffusion characteristics, and propose an artificial neural networks based framework to identify the source by classifying the diffusion characteristics. The feasibility and effectiveness of the proposed framework are validated on a series of synthetic and real networks.

Materials and methods

Problem description and overview

A network the diffusion process taking place in is modelled as a finite and undirected graph $\mathcal{G} = \{V, E, \boldsymbol{\theta}\}$, where V and E represent the nodes set and edges set, respectively. $\boldsymbol{\theta} = \{\theta_{vu}\}$, θ_{vu} is the random propagation delay associated with an edge vu, $vu \in E$. Generally, $\mathcal{G}$ is assumed to be known. We consider that the $\{\theta_{vu}\}$ associated with E are independent and identically distribution (I.I.D) random variables.

Diffusion model. Assuming that the diffusion process taking place in $\mathcal{G}$ follows a simple diffusion model that is similar to reference [50]. At time t, each node $v \in V$ is only in one of the two states: (*i*) informed, if it has received the information from any one neighbour, or (*ii*)

ignorant, if it has not been informed so far. Any node v is equally likely to be the source. The diffusion process is initiated by a single source s^* at unknown start time, all nodes are ignorant except for s^* is informed. Let $\mathcal{N}(v)$ denote the neighbour(s) of v. Suppose v is in the ignorant state, and receives the information for the first time from one informed neighbour w, thus becoming informed at time t_v. Then, v will attempt to retransmit the information to all its other neighbours along the edges, so that each neighbour $u \in \mathcal{N}(v) \setminus w$ receives the information with success probability β at time $t_v + \theta_{vu}$. If there are two or more informed neighbours having a same propagation delay to u, u can be informed by only one neighbour. Once the diffusion process is terminated, a network snapshot, denoted by $\mathcal{S}$, will be generated.

For an arbitrary $\mathcal{G} = \{V, E, \boldsymbol{\theta}\}$, with the diffusion model introduced above, a network snapshot $\mathcal{S}$ is generated. Generally, only a part of nodes state in $\mathcal{S}$ can be observed, we call these nodes observers, denoted by $\mathcal{O}$. The observations made by $\mathcal{O}$ provide two types of information [21, 50]: (*i*) the direction in which information arrives to observers and, (*ii*) the timing at which the information arrives to observers. Obviously, the two types of information recorded in $\mathcal{O}$ show the true details of the diffusion process, which have been proved to be helpful in accurately identifying the diffusion source [21–26]. In this paper, the two types of information are defined as diffusion characteristics. The purpose is to find the diffusion source s^* from $\mathcal{S}$ by utilizing the diffusion characteristics recorded in $\mathcal{O}$. We propose a neural networks based diffusion characteristics classification framework (NN-DCCF) to identify the diffusion source, by which the identification of source is converted into a classification problem. NN-DCCF is composed of the following three stages.

1. By selecting vital nodes and extending their neighbours in a given $\mathcal{G}$, we build $\mathcal{O}$. Then, for a $\mathcal{S}$, by utilizing the diffusion characteristics recorded in $\mathcal{O}$, we construct network snapshot feature, denoted by $\mathcal{F}(\mathcal{S})$.
2. We propose a graph LSTM auto-encoder (GLSTM-AE). By using GLSTM-AE, $\mathcal{F}(\mathcal{S})$ is represented as low-dimension vectors, denoted by $\boldsymbol{\mathcal{R}}(S)$.
3. We propose a source classification neural network (SCNN) to estimate the diffusion source by classifying $\boldsymbol{\mathcal{R}}(S)$.

The overview of NN-DCCF is shown in Fig 1. Frequently used notations are summarized in Table 1.

Stage 1: Constructing network snapshot feature

To utilize the diffusion characteristics of observers to construct network snapshot feature, the observers set $\mathcal{O}$ is built with the following strategy. Given a $\mathcal{G}$, we first rank the importance of nodes by a vital nodes identification methods [51]. Next, with the ranking results, we select the most important K nodes as vital nodes. Then, for each vital node, we extend its neighbours within h hops distance. Further, each vital node and its extended neighbours are combined to form an observation area. $\mathcal{G}$ contains K observation areas $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_K$, $\mathcal{A} = \{\mathcal{A}_i\}_{i=1}^{K}$, $\mathcal{O} = \{\mathcal{A}_1 \cup \mathcal{A}_2 \cup \cdots \cup \mathcal{A}_K\}$. o denotes an unique observer in $\mathcal{O}$. When the diffusion process occurs on $\mathcal{G}$ and generates $\mathcal{S}$, by utilizing the diffusion characteristics, i.e., the diffusion direction and time information, recorded in each $o \in \mathcal{O}$, we construct the network snapshot feature $\mathcal{F}(S)$. Here, we set $K = |\mathcal{A}| = |\mathcal{F}(S)|$. The procedure for constructing $\mathcal{F}(S)$ is summarised in Algorithm 1.

Algorithm 1 Network snapshot feature constructing algorithm

Input: $\mathcal{G}$, $\mathcal{A}$ and $\mathcal{S}$

Output: $\mathcal{F}(S)$

initialize an empty $\mathcal{F}(S) = \{\mathcal{F}(S)_i\}_{i=1}^{|\mathcal{A}|}$

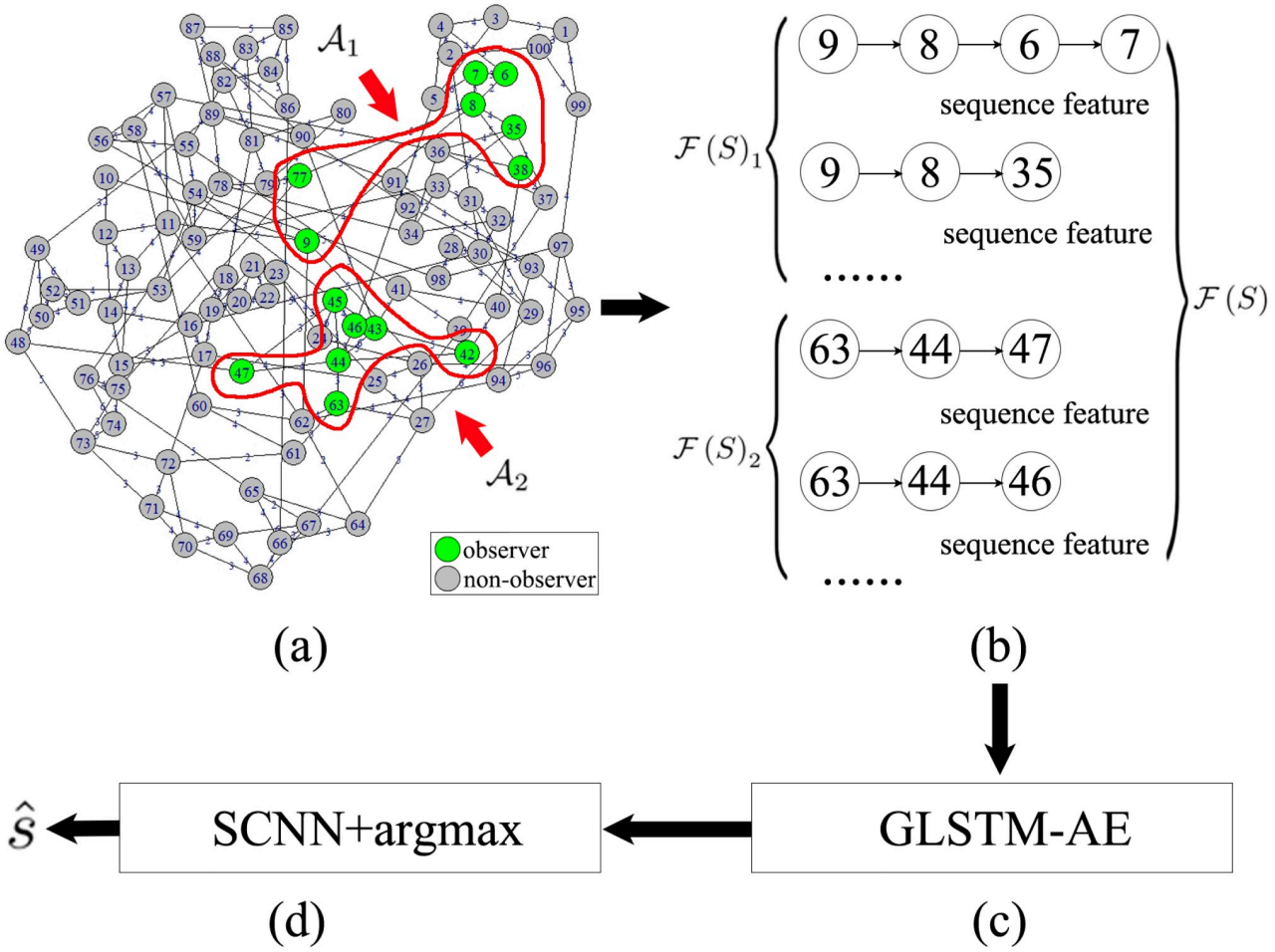

Fig 1. Overview of NN-DCCF. (a) The vital nodes selected by degree centrality [51] include node 8 and node 44. Observation areas set $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2\}$. $\mathcal{A}_1 = \{6, 7, 8, 9, 35, 38, 77\}$. $\mathcal{A}_1$ consists of node 8 and its neighbours within 1 hop distance. $\mathcal{A}_2 = \{42, 43, 44, 45, 46, 47, 63\}$. $\mathcal{A}_2$ consists of node 44 and its neighbours within 1 hop distance. $\mathcal{O} = \{\mathcal{A}_1 \cup \mathcal{A}_2\}$. (b) $\mathcal{F}(\mathcal{S}) = \{\mathcal{F}(\mathcal{S})_1, \mathcal{F}(\mathcal{S})_2\}$. $\mathcal{F}(\mathcal{S})_1$ is composed of the sequence features constructed with the diffusion characteristics recorded in the observers of $\mathcal{A}_1$. $\mathcal{F}(\mathcal{S})_2$ is composed of the sequence features constructed with the diffusion characteristics recorded in the observers of $\mathcal{A}_2$. (c) By using GLSTM-AE, $\mathcal{F}(\mathcal{S})$ is converted into low-dimension representation vectors, i.e. $\mathcal{R}(S)$. (d) With $\mathcal{R}(S)$ as the input of SCNN, we can estimate the diffusion source.

https://doi.org/10.1371/journal.pone.0285563.g001

2: sort all $\mathcal{A}_i$ in $\mathcal{A}$ according to the average informed time of $\mathcal{A}_i$
3: **for** each $\mathcal{A}_i$ in $\mathcal{A}$ **do**
4: **for** each $o \in \mathcal{A}_i$ **do**
5: initialize an empty *seq* to record a single sequence feature
6: set current node $c = o$
7: **while** $c \in \mathcal{A}_i$ **do**
8: **if** c is in the informed state **then**
9: add c into *seq*
10: get next node n according to the diffusion direction information recorded in c and set $c = n$
11: **end if**
12: **end while**
13: reverse the nodes order in *seq*
14: **if** $1 < |seq| \leq l_{max}$ **then**
15: add *seq* into $\mathcal{F}(S)_i$

Table 1. Notation summarization.

Notation	Definition
$\mathcal{G}$	the topological graph of network
V	the nodes set in $\mathcal{G}$
E	the edges set in $\mathcal{G}$
o	observer
$\mathcal{O}$	observers set
$\boldsymbol{\theta}$	propagation delay set associated with E
$\mathcal{S}$	network snapshot
s^*	diffusion source
β	propagation rate of diffusion model
K	the number of vital nodes
$\mathcal{A}$	observation areas set
$\mathcal{F}(\mathcal{S})$	network snapshot feature
$\boldsymbol{\mathcal{R}}(S)$	the representation vectors of $\mathcal{F}(\mathcal{S})$
l	the length of sequence feature
$l_{\max}$	the max length of sequence feature
η	the number of sequence features
$\lvert\cdot\rvert$	the number of elements

https://doi.org/10.1371/journal.pone.0285563.t001

```
    else if |seq| > l_max then
      remove the last |seq| - l_max nodes from seq
      add seq into F(S)_i
    end if
  end for
end for
for each F(S)_i in F(S) do
  remove duplicated sequence features from F(S)_i
  sort all sequence features in F(S)_i according to their length
  if |F(S)_i| > η then
    remove the last |F(S)_i| − η sequence features from F(S)_i
  end if
end for
```

In Algorithm 1, the inputs are the topology of $\mathcal{G}$, observation areas set $\mathcal{A}$ and network snapshot $\mathcal{S}$. The average informed time of $\mathcal{A}_i$ in step 2 is the average of the diffusion time information recorded in $o \in \mathcal{A}_i$. Steps 4–20 are used for constructing the sequence features in $\mathcal{F}(S)_i$ by traversing each $o \in \mathcal{A}_i$. Here, steps 7–12 are used to generate a single sequence feature, denoted by *seq*. A single *seq* is a basic unit for constructing $\mathcal{F}(S)$. Obviously, generating a single *seq* depends on the diffusion direction information of observers. Step 13 is used to reverse the order of current *seq*. Steps 14–19 are used to add the *seq* into $\mathcal{F}(S)_i$, where, $2 \leq |seq| \leq l_{\max}$. Further, from step 3 to step 21, the sequence features in each $\mathcal{F}(S)_i$ are constructed, then, we get $\mathcal{F}(S)$. Steps 22–28 are used to remove the redundant sequence features and limit the size of $\mathcal{F}(S)$. A schematic to obtain $\mathcal{F}(S)$ by using Algorithm 1 is shown in Fig 1(a) and 1(b).

Stage 2: GLSTM-AE based network snapshot feature representation

From Algorithm 1, we know that each sequence feature, termed as *seq*, in $\mathcal{F}(S)$ consists of several ordered informed nodes. Therefore, the *seq* is a type of sequential data. Inspired by the idea that the long short-term memory (S1 File) is a powerful tools for modelling sequential data [52–54], we use the LSTM networks to learn the representation of *seq*. However, the *seq* is

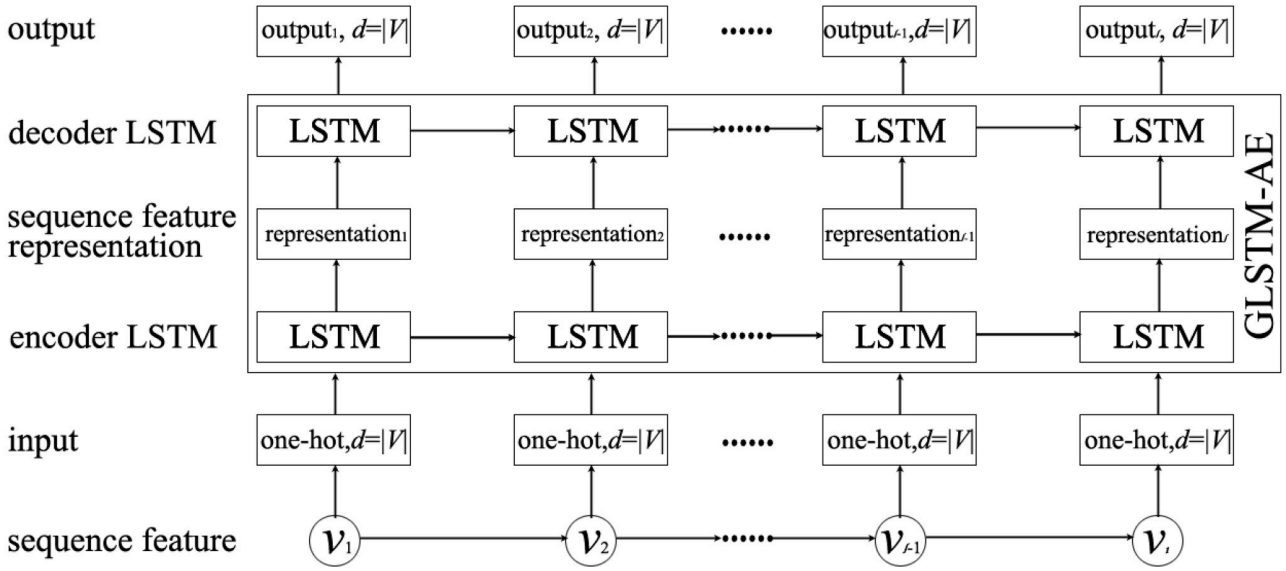

Fig 2. The structure of GLSTM-AE, where, *d* denotes the dimension of a vector.

https://doi.org/10.1371/journal.pone.0285563.g002

different from traditional sequential data since it is composed of ordered informed nodes. Further, we propose a graph LSTM auto-encoder (GLSTM-AE) to learn the low-dimension representation of *seq*. A GLSTM-AE consists of two LSTMs, the encoder LSTM and the decoder LSTM, as shown in Fig 2. GLSTM-AE works as follows. For an arbitrary *seq*, each node in *seq* is represented as an one-hot vector with dimension $|V|$. The input to GLSTM-AE is the one-hot representation of *seq*. The output of the encoder LSTM after the last input has been read is low-dimension representations of the one-hot vectors of *seq*, denoted by $\boldsymbol{r}$, $r \in \mathbf{R}^{|seq| \times d_r}$, where, d_r denotes the representation dimension. $\boldsymbol{r}$ is the representation result we obtained from the GLSTM-AE. The decoder LSTM reconstruct back the input from $\boldsymbol{r}$. The target of GLSTM-AE is same as the input.

Obviously, it is necessary to train GLSTM-AE before it is applied to learn the representations of the sequence features in $\mathcal{F}(S)$. A simple way to obtain the training data of GLSTM-AE is generating sequence features with fixed length from $\mathcal{G}$.

Because the mean squared error (MSE) loss is commonly used for the regression task, it is suitable for the task of GLSTM-AE. Therefore, we adopt the MSE loss as the loss function of GLSTM-AE, which is described as follows.

$$Loss_{GLSTM-AE} = MSELoss(Y, Y^*) \tag{1}$$

where, Y denotes the output of the decoder LSTM in GLSTM-AE, Y^* denotes the one-hot representation of *seq*.

Then, with the trained GLSTM-AE, we get the low-dimension representation of $\mathcal{F}(S)$, denoted by $\boldsymbol{\mathcal{R}}(S)$. This process is summarised in Algorithm 2.

Algorithm 2 Network snapshot feature representation algorithm

Input: $\mathcal{F}(S)$

Output: $\boldsymbol{\mathcal{R}}(S)$

initialize an empty $\boldsymbol{\mathcal{R}}(S)$, $\boldsymbol{\mathcal{R}}(S) \in \mathbf{R}^{K \times \eta \times l_{\max} \times d_r}$

set $\boldsymbol{p}_1 = [0] \in \mathbf{R}^{d_r}$

set $\boldsymbol{p}_\eta = [0] \in \mathbf{R}^{l_{\max} \times d_r}$

```
4: for each F(S)_i in F(S) do
5:   for each seq in F(S)_i do
6:     input=one-hot(seq), input ∈ R^{|seq|×|V|}
7:     r = GLSTM-AE (input), r ∈ R^{|seq|×d_r}
8:     if |seq| < l_max then
9:       k = l_max − |seq|
10:       pad r with p_1 for k times
11:     end if
12:     add r into R(S)_i
13:   end for
14:   if |F(S)_i| < η then
15:     k = η − |F(S)_i|
16:     pad R_i(S) with p_η for k times
17:   end if
18: end for
```

In Algorithm 2, the input is the network snapshot feature $\mathcal{F}(S)$. The one-hot(·) function in step 6 is to get the one-hot representation of current *seq*. In step 7, the representation result $\boldsymbol{r}$ of *seq* is obtained by using the trained GLSTM-AE. Further, we set $\boldsymbol{r} \in \mathbf{R}^{l_{\max} \times d_r}$ by steps 8–11, and set $\boldsymbol{\mathcal{R}}_i \in \mathbf{R}^{\eta \times l_{\max} \times d_r}$ by steps 14–17.

Stage 3: Identify the diffusion source with SCNN

With Algorithm 2, we get the representation of $\mathcal{F}(S)$, i.e. $\boldsymbol{\mathcal{R}}(S)$. In this section, with $\boldsymbol{\mathcal{R}}(S)$ as input, we propose a source classification neural network (SCNN) to identify the diffusion source by classifying $\boldsymbol{\mathcal{R}}(S)$. SCNN is mainly composed of two fully connected layers. To get convergence faster, we add a normalization layer. The structure of SCNN is shown in Fig 3, where, the LogSoftmax is used for multi-class classification.

SCNN also requires to be trained before it is applied to identify the diffusion source. The training data of SCNN can be generated by Algorithm 3.

Algorithm 3 SCNN training data generating algorithm

```
Input: G and A
Output: training data collector C
1: specify the number of loops N
2: initialize an empty training data collector C
3: set β = {β_i}_{i=1}^M, β_i ∈ (0, 1), ∀i, j ∈ [1, M], β_i ≠ β_j
4: while N > 0 do
5:   for β_i ∈ β do
6:     for each node v ∈ V do
7:       generate S by running the diffusion model (see Diffusion
model) on G with v as diffusion source and β_i as propagation rate
8:       generate F(S) by Algorithm 1
9:       construct R(S) corresponding to F(S) by Algorithm 2
10:       add a training data (R(S), one-hot(v)) into C
11:     end for
12:   end for
13:   N = N − 1
14: end while
```

In Algorithm 3, the inputs are the topology of $\mathcal{G}$ and observation areas set $\mathcal{A}$. From step 7 to step 10, given a node v and a propagation rate β, a single training data can be generated, which is composed of the $\boldsymbol{\mathcal{R}}(S)$ and the one-hot representation of v. Obviously, the SCNN training dataset size is $N \cdot |\beta| \cdot |V|$.

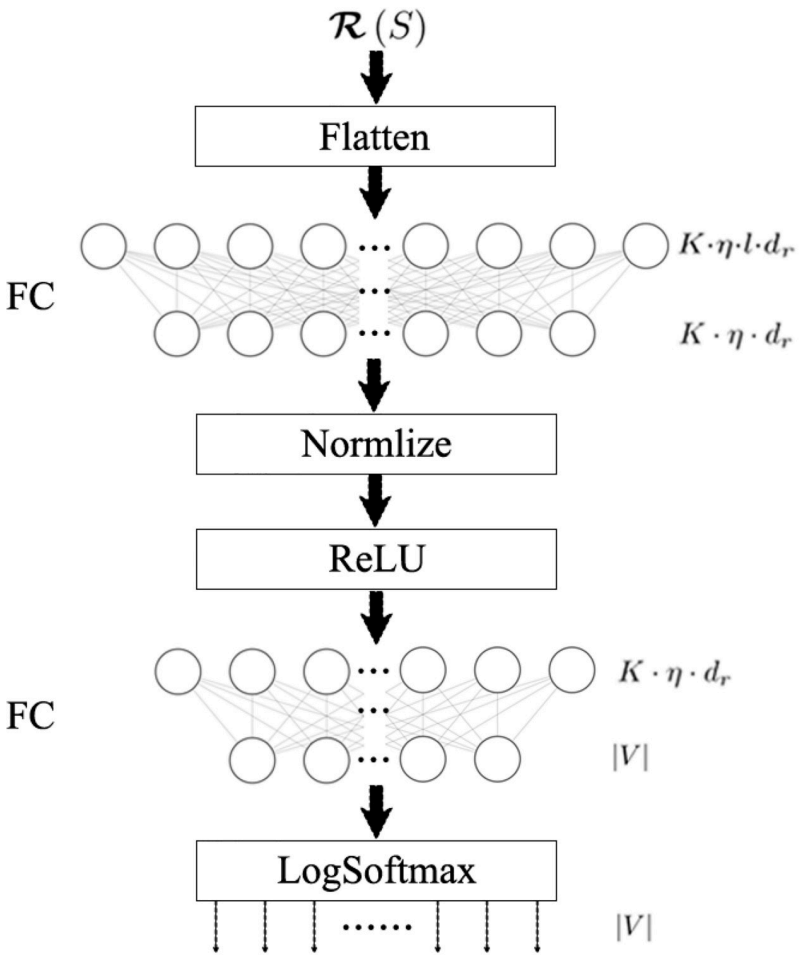

Fig 3. The structure of SCNN.

https://doi.org/10.1371/journal.pone.0285563.g003

Because the cross entropy loss is mainly used for classification, we adopt the cross entropy loss as the loss function of SCNN, which is described as follows.

$$Loss_{SCNN} = CrossEntropyLoss(Z, Z^*) \tag{2}$$

where, Z denotes the estimated diffusion source obtained by SCNN, Z^* denotes the one-hot representation of true diffusion source.

Finally, by combining with the trained SCNN, the algorithm corresponding to NN-DCCF is summarised as Algorithm 4.

Algorithm 4 Diffusion source identification algorithm

Input: $\mathcal{G}$, $\mathcal{A}$ and $\mathcal{S}$
Output: $\hat{s}$

1: generate $\mathcal{F}(\mathcal{S})$ according Algorithm 1
2: construct $\boldsymbol{\mathcal{R}}(\mathcal{S})$ corresponding to $\mathcal{F}(\mathcal{S})$ according to Algorithm 2
3: ***output*** = SCNN $(\boldsymbol{\mathcal{R}}(\boldsymbol{S}))$, ***output*** $\in \mathbf{R}^{|V|}$
4: $\hat{s} = \arg\max(output)$

Results

Main experimental environment

Hardware: Dell R740 with 2 Intel(R) Xeon(R) gold 6254 CPU, 1 TB RAM, 1 NVIDIA Tesla V100S GPU with 32 GB GPU memory. Software: Python 3.8.10 + PyTorch 1.10.2 + CUDA 10.2.

Methods for comparison

Essentially, the proposed NN-DCCF is an observers based method. To validate its feasibility and effectiveness, three existing state-of-the-art observers based methods are selected for comparison, including time-reversal backward spreading (TRBS) algorithm [24], sequential neighbour filtering (SNF) algorithm [48] and estimated propagation delay (EPD) algorithm [49].

Datasets

We compare the four diffusion source identification methods on a series of synthetic and real networks. The synthetic networks include scale-free (BA) [55] model and small-world (WS) [56] model. The parameters for generating synthetic networks are summarised in Tables 2 and 3. The real networks are of different types, including NetworkScience (https://networkrepository.com/ca-netscience.php) [57], Euroroads (https://networkrepository.com/subelj-euroroad.php) [57], Email (https://networkrepository.com/email-univ.php) [57] and Blogs (https://doi.org/10.1007/978–3-642-01206-8_5) [58]. The topological properties of all networks are summarised in Table 4.

Evaluation metrics

The performance of diffusion source identification methods are commonly evaluated with two metrics [5, 21, 25, 27, 34], including the precision and average error distance. The precision focuses on evaluating the capability of a method in precise identification (i.e. the proportion of 0 error hop). For each network, we randomly select 100 nodes as test seeds. For the precision, the higher the value is, the better the algorithm is. For the average error hop, the smaller the value is, the better the algorithm is.

Table 2. The parameters for generating BA models.

Network	$\|V\|$	M	*power*
BA model (1)	400	2	1.3
BA model (2)	1000	2	1.3

M: the number of outgoing edges generated for each node.
power: the power of the preferential attachment [55].

https://doi.org/10.1371/journal.pone.0285563.t002

Table 3. The parameters for generating WS models.

Network	$\|V\|$	*Nei*	p
WS model (1)	400	2	0.1
WS model (2)	1000	2	0.1

Nei: the size of the neighbourhood for each node.
p: the rewiring probability [56].

https://doi.org/10.1371/journal.pone.0285563.t003

Table 4. The topological properties of networks.

Network	$\|V\|$	$\|E\|$	$\langle k\rangle$	A	H	APL
BA model (1)	400	797	3.99	−0.166	2.64	3.55
BA model (2)	1000	1997	3.99	−0.131	8.45	3.41
WS model (1)	400	800	4.00	−0.014	1.05	5.87
WS model (2)	1000	2000	4.00	0.038	1.05	6.85
NetworkScience	379	914	4.82	−0.082	1.66	6.04
Euroroads (LCC)	1039	1305	2.51	0.090	1.23	18.4
Email	1133	5451	9.62	0.078	1.94	3.61
Blogs	3982	6803	3.42	−0.133	4.04	6.25

$\langle k\rangle$: average degree.

A: the assortative coefficient [59].

H: the degree heterogeneity [60]

$H = \frac{\langle k^2\rangle}{\langle k\rangle^2}$.

APL: the average path length (the number of edges).

LCC: largest connected component.

https://doi.org/10.1371/journal.pone.0285563.t004

Parameters setting

For an arbitrary $\mathcal{G} = \{V, E, \boldsymbol{\theta}\}$, we assume that $\boldsymbol{\theta}$ are Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ [24, 49], μ and σ^2 are known [50], here, we set $\mu/\sigma = 4$ [21, 50]. We assume that the diffusion process on networks follows the diffusion model introduced in Diffusion model. To investigate the performance of NN-DCCF under different propagation rates, we set relatively larger range for β, $\beta \in [0.1, 0.9]$. The diffusion process is terminated when there is no ignorant node.

How to select a suitable observers placement strategy may depends on the topology of a network [61]. Although lots of methods [51] can be used to select $\mathcal{O}$, sometimes there maybe no significant difference between the placement strategies for the performance of source identification [62]. In this paper, $\mathcal{O}$ is selected by the strategy introduced in Section Stage 1. Here, the K vital nodes in $\mathcal{G}$ are selected by the degree centrality (DC) [51] (due to the simplicity and efficiency of DC). For each network, we select 1% nodes as vital nodes. Then, by extending the neighbours within 1 hop distance of the vital nodes, we get $\mathcal{A}$ and $\mathcal{O}$ in $\mathcal{G}$, the details are shown in Table 5. Other general parameters are also summarised in Table 5. All the four compared methods adopt the same $\mathcal{O}$ to identify the diffusion source.

Table 5. General parameters setting.

Network	$\|V\|$	K	$\|\mathcal{A}\|$	$\|\mathcal{O}\|$	$l_{\max}$	η
BA model (1)	400	4	4	144	4	8
BA model (2)	1000	10	10	594	4	8
WS model (1)	400	4	4	32	4	8
WS model (2)	1000	10	10	76	4	8
NetworkScience	379	4	4	73	4	10
Euroroads	1039	11	11	88	4	6
Email	1133	12	12	355	4	20
Blogs	3982	40	40	1646	4	8

$\eta = 2 \cdot \lceil \langle k\rangle \rceil$, $\langle k\rangle$ can be found in Table 4.

https://doi.org/10.1371/journal.pone.0285563.t005

Table 6. The parameters set of GLSTM-AE.

parameter	dimension
d_I	$\|V\|$
d_r	8, $\|V\|<1000$
	16, $\|V\|\geq 1000$
d_O	$\|V\|$

d_I: input dimension of GLSTM-AE.

d_r: the dimension of representation result obtained by GLSTM-AE.

d_O: output dimension of GLSTM-AE.

https://doi.org/10.1371/journal.pone.0285563.t006

Table 7. The training dataset size of GLSTM-AE and SCNN on different networks.

Network	GLSTM-AE	SCNN
BA model (1)	115282	28800
BA model (2)	991382	54000
WS model (1)	21896	28800
WS model (2)	54784	54000
NetworkScience	92902	27288
Euroroads	19752	56106
Email	3714528	101970
Blogs	1253118	143352

https://doi.org/10.1371/journal.pone.0285563.t007

The parameters set of GLSTM-AE are summarised in Table 6. Meanwhile, we generate the training dataset of GLSTM-AE for each network with the simple method introduced in Section Stage 2. To emphasize the local structure, we set $l \in [2, 4]$. The training dataset size of GLSTM-AE on different networks are shown in Table 7. The training parameters set for GLSTM-AE on different networks are summarised in Table 8. Because the purpose is to identify the diffusion source, we show the accuracy of GLSTM-AE by the results of source identification, which can be found in Figs 4–11 and Table 10.

Table 8. The training parameters set of GLSTM-AE and SCNN on different networks.

Network	GLSTM-AE			SCNN		
	BS	LR	epochs	BS	LR	epochs
BA model (1)	256	0.01	40	256	0.001	20
BA model (2)	1024	0.01	10	1024	0.0001	20
WS model (1)	256	0.03	40	256	0.0002	20
WS model (2)	1024	0.02	40	1024	0.0001	20
NetworkScience	256	0.01	40	256	0.0001	20
Euroroads	1024	0.02	40	512	0.0001	20
Email	2048	0.02	10	2048	0.0002	20
Blogs	2048	0.02	10	2048	0.0002	20

BS: batch size.

LR: learning rate.

https://doi.org/10.1371/journal.pone.0285563.t008

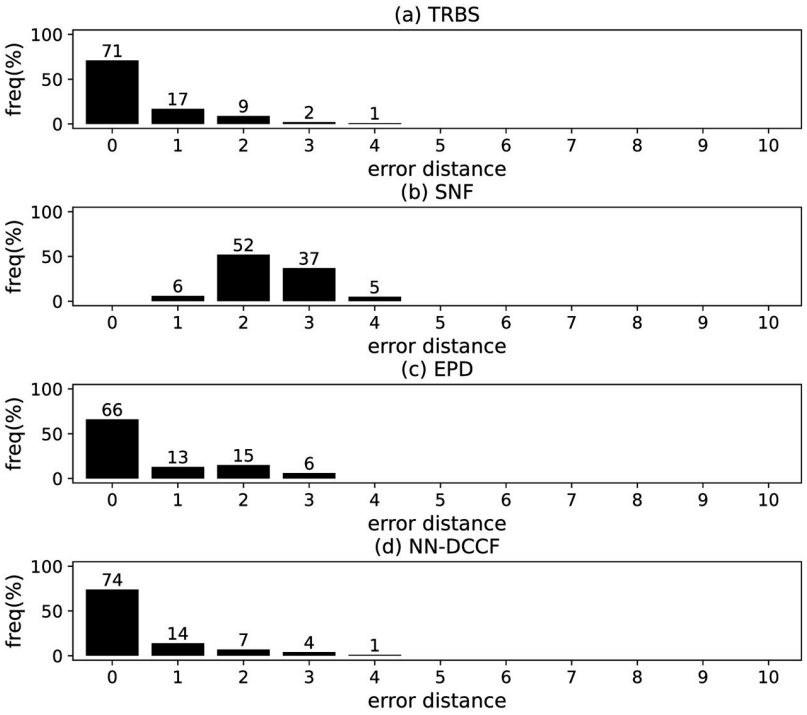

Fig 4. The error distance of TRBS, SNF, EPD and NN-DCCF methods on BA model (1).

https://doi.org/10.1371/journal.pone.0285563.g004

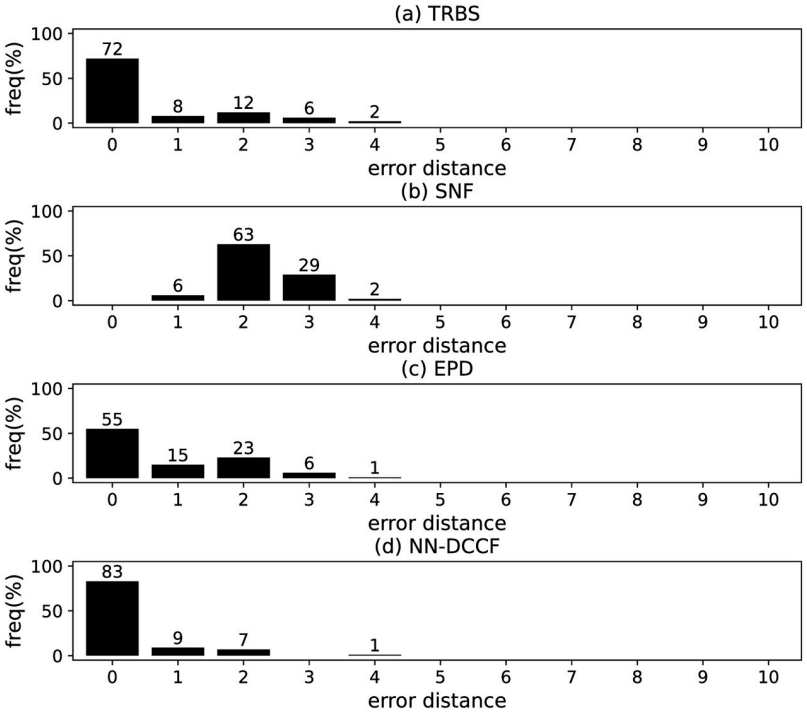

Fig 5. The error distance of TRBS, SNF, EPD and NN-DCCF methods on BA model (2).

https://doi.org/10.1371/journal.pone.0285563.g005

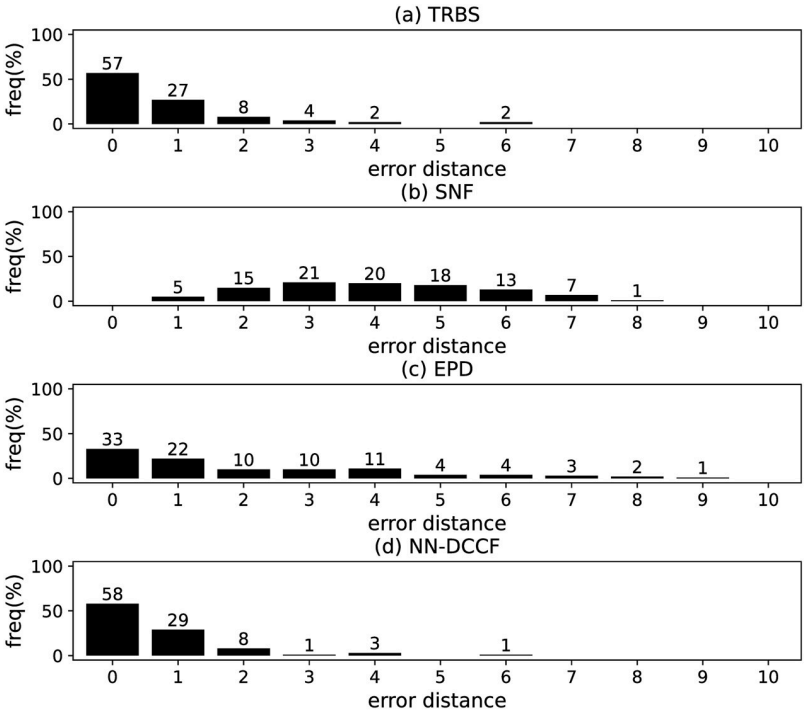

Fig 6. The error distance of TRBS, SNF, EPD and NN-DCCF methods on WS model (1).

https://doi.org/10.1371/journal.pone.0285563.g006

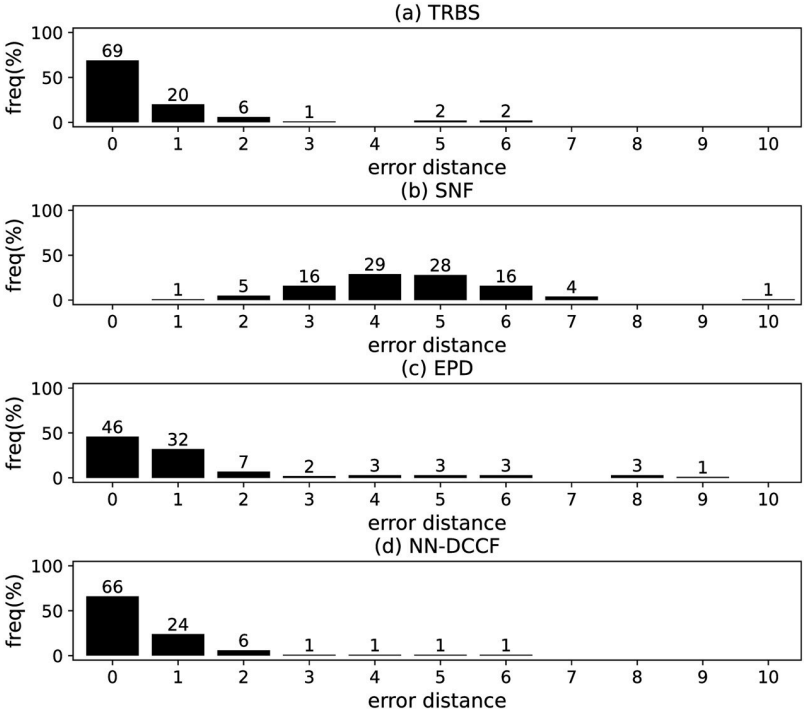

Fig 7. The error distance of TRBS, SNF, EPD and NN-DCCF methods on WS model (2).

https://doi.org/10.1371/journal.pone.0285563.g007

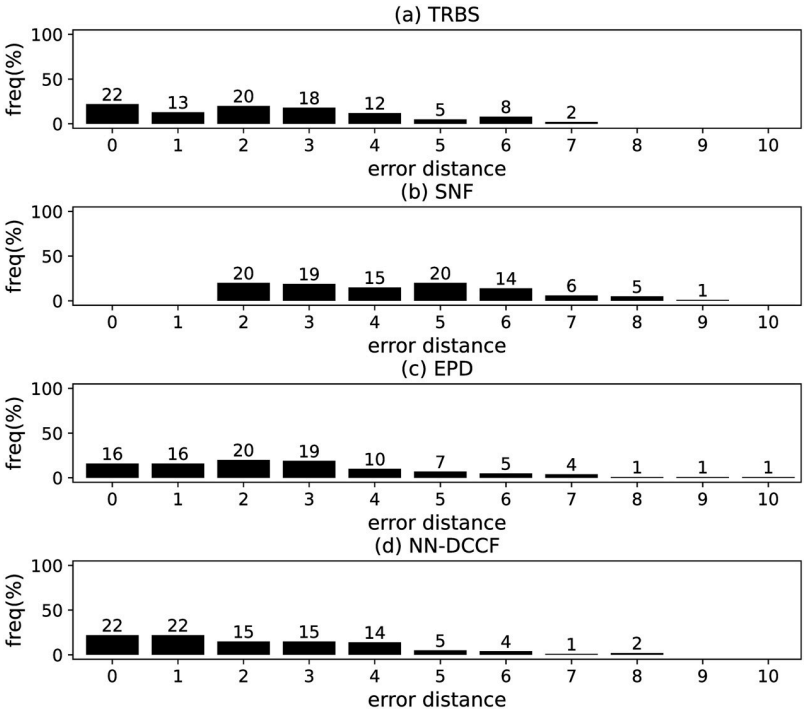

Fig 8. The error distance of TRBS, SNF, EPD and NN-DCCF methods on NetworkScience network.

https://doi.org/10.1371/journal.pone.0285563.g008

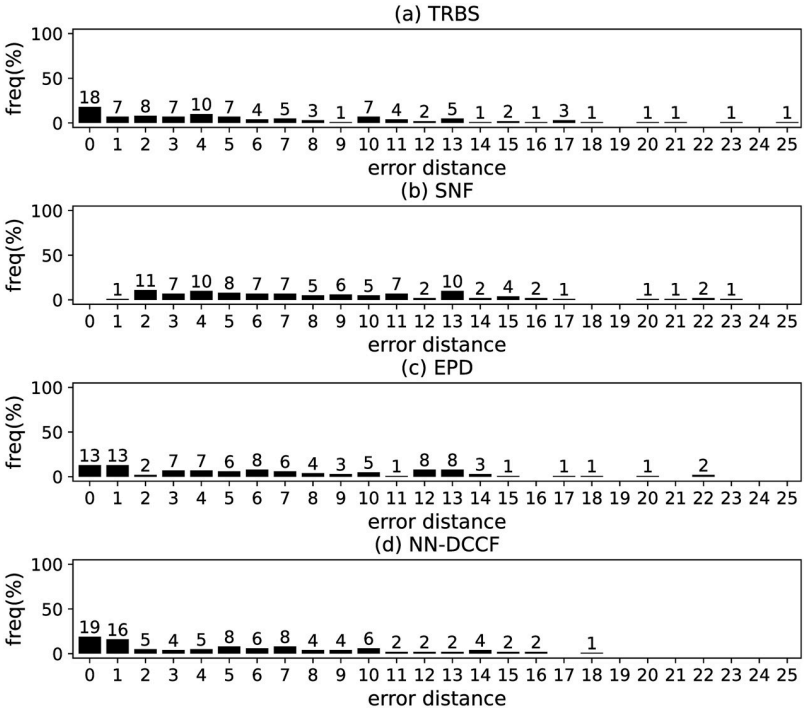

Fig 9. The error distance of TRBS, SNF, EPD and NN-DCCF methods on Euroroads network.

https://doi.org/10.1371/journal.pone.0285563.g009

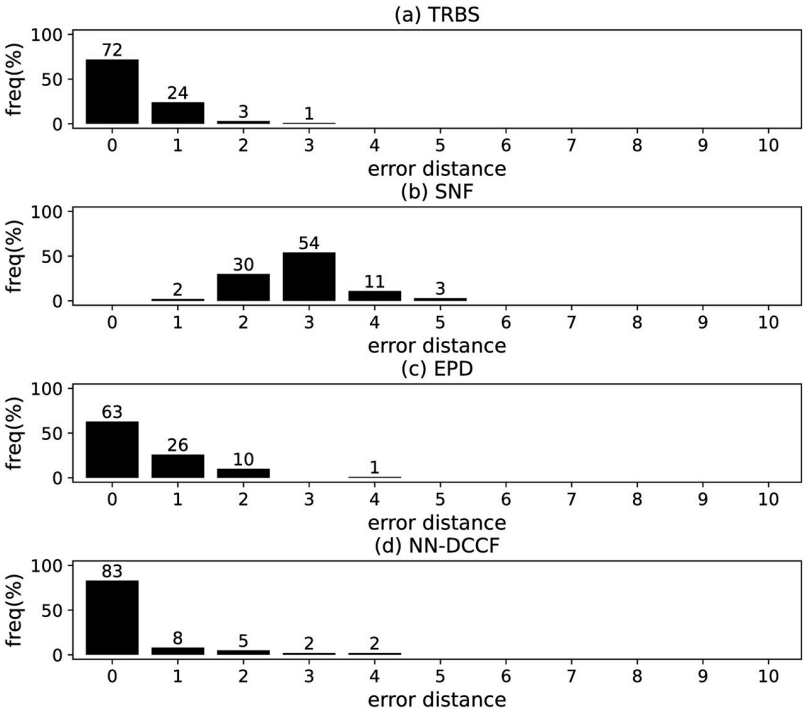

Fig 10. The error distance of TRBS, SNF, EPD and NN-DCCF methods on Email network.

https://doi.org/10.1371/journal.pone.0285563.g010

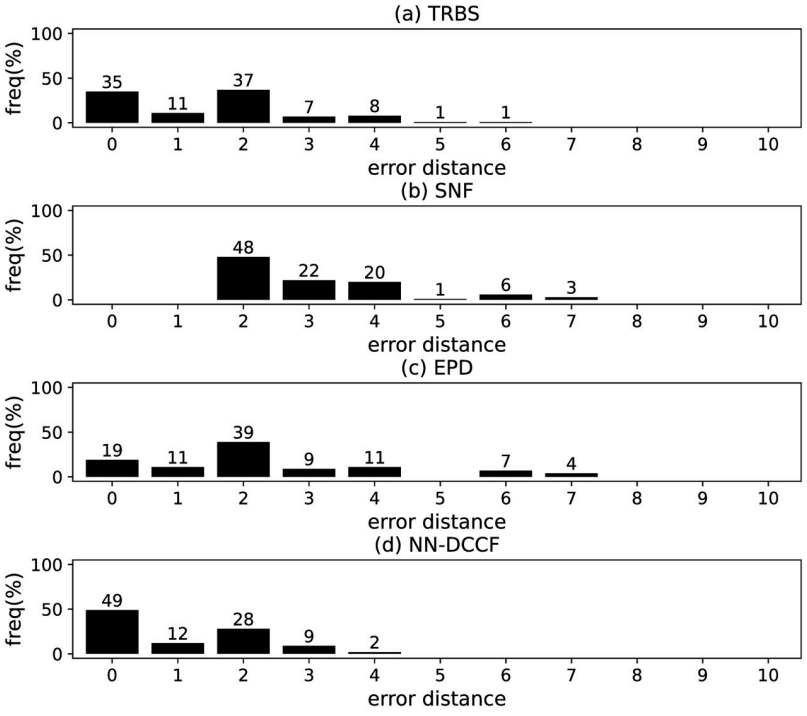

Fig 11. The error distance of TRBS, SNF, EPD and NN-DCCF methods on Blogs network.

https://doi.org/10.1371/journal.pone.0285563.g011

Table 9. The parameters set of SCNN.

Parameter	dimension
The input of first FC layer	$K \cdot \eta \cdot l \cdot d_r$
The output of first FC layer	$K \cdot \eta \cdot d_r$
The input of second FC layer	$K \cdot \eta \cdot d_r$
The output of second FC layer	$\|V\|$
LogSoftmax layer	$\|V\|$

https://doi.org/10.1371/journal.pone.0285563.t009

Table 10. The average error distance of TRBS, SNF, EPD and NN-DCCF on different networks.

network	TRBS	SNF	EPD	NN-DCCF
BA model(1)	0.45	2.41	0.61	**0.44**
BA model(2)	0.58	2.27	0.83	**0.27**
WS model(1)	0.75	4.03	2.06	**0.66**
WS model(2)	0.57	4.49	1.30	**0.54**
NetworkScience	2.42	4.32	2.73	**2.25**
Euroroads	6.32	8.35	6.62	**5.26**
Email	0.33	2.83	0.50	**0.32**
Blogs	1.49	3.04	2.30	**1.03**

https://doi.org/10.1371/journal.pone.0285563.t010

The parameters set of SCNN are summarised in Table 9. Further, for each network, we generate the training dataset of SCNN by Algorithm 3. The training dataset size of SCNN on different networks are shown in Table 7. The training parameters set for SCNN are summarised in Table 8. In SCNN, we adopt the batch normalization as the normalization layer [19, 20]. Since the purpose is to identify the diffusion source, we validate the performance of SCNN by the results of source identification, which can be found in Figs 4–11 and Table 10.

Experimental results and discussion. Figs 4–11 show the error distance of the four methods on different networks. Table 10 shows the average error distance of the four methods. From Figs 4 to 11, we can see that the precisions (i.e. the proportion of 0 error hop) exposed by NN-DCCF on the eight networks are 74%, 83%, 58%, 66%, 22%, 19%, 83% and 49%, respectively. Obviously, except for WS model (2), NN-DCCF exposes the best performance in precision. On WS model (2), the precision of NN-DCCF is only inferior to the TRBS, and superior to other two methods. From Table 10, we know that the NN-DCCF is superior to other three methods in the average error distance on all networks. Therefore, the NN-DCCF is a feasible and effective method in accurately identifying the diffusion source. Additionally, from Table 4, we know that the eight networks are different in their topological properties, which indicates that NN-DCCF could effectively identify the source on different types of networks by simply modifying the training parameters. Therefore, the NN-DCCF is a general source identification framework.

Conclusion

This paper defines the diffusion direction and time information of observers as diffusion characteristics, and develops a NN-DCCF to identify the diffusion source by classifying the diffusion characteristics. Firs, we utilize the diffusion characteristics to construct network snapshot feature. Then, we propose a GLSTM-AE by which the network snapshot feature is represented as low-dimension vectors. Further, we propose a SCNN to identify the diffusion source. By

using NN-DCCF, the identification of diffusion source is converted into a classification problem. The feasibility and effectiveness of NN-DCCF are validated by the experimental results on a series of synthetic and real networks. In the future work, we will generalize the NN-DCCF to the case of multi-source.

Supporting information

S1 File. Long short-term memory (LSTM).
(PDF)

Author Contributions

Conceptualization: Fan Yang, Yabing Yao.

Methodology: Fan Yang, Jingxian Liu, Ruisheng Zhang.

Software: Fan Yang, Jingxian Liu.

Supervision: Jingxian Liu, Ruisheng Zhang, Yabing Yao.

Validation: Fan Yang, Yabing Yao.

Writing – original draft: Fan Yang, Ruisheng Zhang, Yabing Yao.

References

1. Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang DU. Complex networks: Structure and dynamics. Physics Reports. 2006; 424(4): 175–308. https://doi.org/10.1016/j.physrep.2005.10.009
2. Chang S, Pierson E, Koh PW, Gerardin J, Redbird B, Grusky D, et al. Mobility network models of COVID-19 explain inequities and inform reopening. Nature. 2021; 589(7840): 82–87. https://doi.org/10.1038/s41586-020-2923-3 PMID: 33171481
3. Zhu L, Yang F, Guan G, Zhang Z. Modeling the dynamics of rumor diffusion over complex networks. Information Sciences. 2021; 562: 240–258. https://doi.org/10.1016/j.ins.2020.12.071
4. Wang Y, Wen S, Xiang Y, Zhou W. Modeling the Propagation of Worms in Networks: A Survey. IEEE Communications Surveys & Tutorials. 2014; 16(2): 942–960. https://doi.org/10.1109/SURV.2013.100913.00195
5. Jiang J, Sheng W, Shui Y, Yang X, Zhou W. Identifying Propagation Sources in Networks: State-of-the-Art and Comparative Studies. IEEE Communications Surveys & Tutorials. 2017; 19(1): 465–481. https://doi.org/10.1109/COMST.2016.2615098
6. Brockmann D, Helbing D. The hidden geometry of complex, network-driven contagion phenomena. Science. 2013; 342(6164): 1337–1342. https://doi.org/10.1126/science.1245200 PMID: 24337289
7. Wang Y, Zhong L, Du J, Gao J, Wang Q. Identifying the shifting sources to predict the dynamics of COVID-19 in the US. Chaos: An Interdisciplinary Journal of Nonlinear Science. 2022; 32(3): 033104. https://doi.org/10.1063/5.0051661
8. Li J, Manitz J, Bertuzzo E, Kolaczyk ED. Sensor-based localization of epidemic sources on human mobility networks. PLoS Computational Biology. 2021; 17(1): e1008545. https://doi.org/10.1371/journal.pcbi.1008545 PMID: 33503024
9. Horn AL, Friedrich H. Locating the source of large-scale outbreaks of foodborne disease. Journal of the Royal Society Interface. 2019; 16(151): 20180624. https://doi.org/10.1098/rsif.2018.0624 PMID: 30958197
10. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE. A survey of deep neural network architectures and their applications. Neurocomputing. 2017; 234: 11–26. https://doi.org/10.1016/j.neucom.2016.12.038
11. Chamberlain B, Rowbottom J, Gorinova MI, Bronstein M, Webb S, Rossi E. GRAND: Graph Neural Diffusion. Proceedings of the 38th International Conference on Machine Learning. 2021; 139: 1407–1418. Available: http://proceedings.mlr.press/v139/chamberlain21a/chamberlain21a.pdf
12. Zhang C, Zhao S, Yang Z, Chen Y. A reliable data-driven state-of-health estimation model for lithium-ion batteries in electric vehicles. Frontiers in Energy Research. 2022; 10. https://doi.org/10.3389/fenrg.2022.1013800

13. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A Comprehensive Survey on Graph Neural Networks. IEEE Transactions on Neural Networks and Learning Systems. 2021; 32(1): 4–24. https://doi.org/10.1109/TNNLS.2020.2978386 PMID: 32217482

14. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. Computational Capabilities of Graph Neural Networks. IEEE Transactions on Neural Networks. 2009; 20(1): 81–102. PMID: 19129034

15. Cui P, Wang X, Pei J, Zhu W. A Survey on Network Embedding. IEEE Transactions on Knowledge and Data Engineering. 2019. 31(5): 833–852. https://doi.org/10.1109/TKDE.2018.2849727

16. Zhang D, Yin J, Zhu X, Zhang C. Network Representation Learning: A Survey. IEEE Transactions on Big Data. 2020; 6: 3–28. https://doi.org/10.1109/TBDATA.2018.2850013

17. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The Graph Neural Network Model. IEEE Transactions on Neural Networks. 2009; 20(1): 61–80. https://doi.org/10.1109/TNN.2008.2005605 PMID: 19068426

18. Kipf TN, Welling M. Semi-Supervised Classification with Graph Convolutional Networks. International Conference on Learning Representations. 2017.

19. Li L, Zhou J, Jiang Y, Huang B. Propagation source identification of infectious diseases with graph convolutional networks. Journal of biomedical informatics. 2021; 116: 103720. https://doi.org/10.1016/j.jbi.2021.103720 PMID: 33640536

20. Dong M, Zheng B, Li G, Li C, Zheng K, Zhou X. Wavefront-Based Multiple Rumor Sources Identification by Multi-Task Learning. IEEE Transactions on Emerging Topics in Computational Intelligence. 2022; 6 (5): 1068–1078. https://doi.org/10.1109/TETCI.2022.3142627

21. Yang F, Yang S, Peng Y, Yao Y, Wang Z, Li H, et al. Locating the propagation source in complex networks with a direction-induced search based Gaussian estimator. Knowledge-Based Systems. 2020; 195: 105674. https://doi.org/10.1016/j.knosys.2020.105674

22. Zhu P, Cheng L, Gao C, Wang Z, Li X. Locating Multi-Sources in Social Networks With a Low Infection Rate. IEEE Transactions on Network Science and Engineering. 2022; 9(3): 1853–1865. https://doi.org/10.1109/TNSE.2022.3153968

23. Cheng L, Li X, Han Z, Luo T, Ma L, Zhu P. Path-based multi-sources localization in multiplex networks. Chaos, Solitons & Fractals. 2022; 159: 112139. https://doi.org/10.1016/j.chaos.2022.112139

24. Shen Z, Cao S, Wang W, Di Z, Stanley HE. Locating the source of diffusion in complex networks by time-reversal backward spreading. Physical Review. E. 2016; 93(3): 032301. https://doi.org/10.1103/PhysRevE.93.032301 PMID: 27078360

25. Tang W, Ji F, Tay WP. Estimating Infection Sources in Networks Using Partial Timestamps. IEEE Transactions on Information Forensics and Security. 2018; 13(12): 3035–3049. https://doi.org/10.1109/TIFS.2018.2837655

26. Hu Z. Wang L, Tang C. Locating the source node of diffusion process in cyber-physical networks via minimum observers. Chaos: An Interdisciplinary Journal of Nonlinear Science. 2019; 29(6): 063117. https://doi.org/10.1063/1.5092772 PMID: 31266325

27. Shah D, Zaman T. Rumors in a Network: Who's the Culprit? IEEE Transactions on Information Theory. 2011; 57(8): 5163–5181. https://doi.org/10.1109/TIT.2011.2158885

28. Luo W, Tay WP, Leng M. Identifying Infection Sources and Regions in Large Networks. IEEE Transactions on Signal Processing. 2013; 61(11): 2850–2865. https://doi.org/10.1109/TSP.2013.2256902

29. Wang Z, Dong W, Zhang W, Tan CW. Rumor Source Detection with Multiple Observations: Fundamental Limits and Algorithms. SIGMETRICS Perform. Eval. Rev. 2014; 42(1): 1–13. https://doi.org/10.1145/2637364.2591993

30. Wang Z, Dong W, Zhang W, Tan CW. Rooting our Rumor Sources in Online Social Networks: The Value of Diversity From Multiple Observations. IEEE Journal of Selected Topics in Signal Processing. 2015; 9(4): 663–677. https://doi.org/10.1109/JSTSP.2015.2389191

31. Zhu K, Ying L. Information Source Detection in the SIR Model: A Sample-Path-Based Approach. IEEE/ACM Transactions on Networking. 2016; 24(1): 408–421. https://doi.org/10.1109/TNET.2014.2364972

32. Zhu K, Ying L. A Robust Information Source Estimator with Sparse Observations. Computational Social Networks. 2014; 1(1): 1–21. https://doi.org/10.1186/s40649-014-0003-2

33. Luo W, Tay WP, Leng M. How to Identify an Infection Source With Limited Observations. IEEE Journal of Selected Topics in Signal Processing. 2014; 8(4): 586–597. https://doi.org/10.1109/JSTSP.2014.2315533

34. Jiang J, Wen S, Yu S, Xiang Y, Zhou W. Rumor Source Identification in Social Networks with Time-Varying Topology. IEEE Transactions on Dependable and Secure Computing. 2018; 15(1): 166–179. https://doi.org/10.1109/TDSC.2016.2522436

35. Lokhov AY, Mézard M, Ohta H, Zdeborová L. Inferring the origin of an epidemic with a dynamic message-passing algorithm. Physical Review E. 2014; 90(1): 012801. https://doi.org/10.1103/PhysRevE.90.012801 PMID: 25122336

36. Altarelli F, Braunstein A, Dall'Asta L, Lage-Castellanos A, Zecchina R. Bayesian inference of epidemics on networks via belief propagation. Physical Review Letters. 2014; 112(11): 118701. https://doi.org/10.1103/PhysRevLett.112.118701 PMID: 24702425

37. Antulov-Fantulin N, Lančić A, Šmuc T, Štefančić H, Šikić M. Identification of Patient Zero in Static and Temporal Networks: Robustness and Limitations. Physical Review Letters. 2015; 114(24): 248701. https://doi.org/10.1103/PhysRevLett.114.248701 PMID: 26197016

38. Yang F, Zhang R, Yao Y, Yuan Y. Locating the propagation source on complex networks with Propagation Centrality algorithm. Knowledge-Based Systems. 2016; 100: 112–123. https://doi.org/10.1016/j.knosys.2016.02.013

39. Zhou J, Jiang Y, Huang B. Source identification of infectious diseases in networks via label ranking. PLoS ONE. 2021; 16(1): e0245344. https://doi.org/10.1371/journal.pone.0245344 PMID: 33444390

40. Chai Y, Wang Y, Zhu L. Information Sources Estimation in Time-Varying Networks. IEEE Transactions on Information Forensics and Security. 2021; PP(99): 2621–2636. https://doi.org/10.1109/TIFS.2021.3050604

41. Jiang J, Wen S, Yu S, Xiang Y, Zhou W. K-Center: An Approach on the Multi-Source Identification of Information Diffusion. IEEE Transactions on Information Forensics and Security. 2015; 10(12): 2616–2626. https://doi.org/10.1109/TIFS.2015.2469256

42. Cai K, Hong X, Lui JCS. Information Spreading Forensics via Sequential Dependent Snapshots. IEEE/ACM Transactions on Networking. 2018; 26(1): 478–491. https://doi.org/10.1109/TNET.2018.2791412

43. Feizi S, Médard M, Quon G, Kellis M, Duffy K. Network Infusion to Infer Information Sources in Networks. IEEE Transactions on Network Science and Engineering. 2019; 6(3): 402–417. https://doi.org/10.1109/TNSE.2018.2854218

44. Chang B, Chen E, Zhu F, Liu Q, Xu T, Wang Z. Maximum a Posteriori Estimation for Information Source Detection. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2020; 50(6): 2242–2256. https://doi.org/10.1109/TSMC.2018.2811410

45. Caputo JG, Hamdi A, Knippel A. Inverse source problem in a forced network. Inverse Problems. 2019; 35(5): 055006. https://doi.org/10.1088/1361-6420/aafcc6

46. Fu L, Shen Z, Wang W, Fan Y, Di Z. Multi-source localization on complex networks with limited observers. EPL. 2016; 113(1): 18006. https://doi.org/10.1209/0295-5075/113/18006

47. Paluch R, Lu X, Suchecki K, Szymański BK, Holyst JA. Fast and accurate detection of spread source in large complex networks, Scientific Reports. 2018; 8(1): 2508. https://doi.org/10.1038/s41598-018-20546-3 PMID: 29410504

48. Wang H, Sun K. Locating source of heterogeneous propagation model by universal algorithm. Europhysics Letters. 2020; 131(4): 48001. https://dx.doi.org/10.1209/0295-5075/131/48001

49. Wang H, Zhang F, Sun K. An algorithm for locating propagation source in complex networks. Physics Letters A. 2021; 393: 127184. https://doi.org/10.1016/j.physleta.2021.127184

50. Pinto PC, T Patrick, V Martin. Locating the source of diffusion in large-scale networks. Physical Review Letters. 2012; 109(6): 068702. https://doi.org/10.1103/PhysRevLett.109.068702 PMID: 23006310

51. Lü L, Chen D, Ren X, Zhang Q, Zhang Y, Zhou T. Vital nodes identification in complex networks. Physics Reports. 2016; 650: 1–63. https://doi.org/10.1016/j.physrep.2016.06.007

52. Sutskever I, Vinyals O, Le QV. Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems. 2014; 27. Available: https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf

53. Srivastava N, Mansimov E, Salakhudinov R. Unsupervised learning of video representations using lstms. International conference on machine learning. 2015. pp. 843–852. Available: http://proceedings.mlr.press/v37/srivastava15.pdf

54. Dai AM, Le QV. Semi-supervised Sequence Learning. Advances in Neural Information Processing Systems. 2015; 28. Available: https://proceedings.neurips.cc/paper/2015/file/7137debd45ae4d0ab9aa953017286b20-Paper.pdf

55. Barabasi AL, Albert R. Emergence of Scaling in Random Networks. Science. 1999; 286(5439): 509–512. https://doi.org/10.1126/science.286.5439.509 PMID: 10521342

56. Watts DJ, Strogatz SH. Collective dynamics of'small-world' networks. Nature. 1998; 393(6684): 440–442. https://doi.org/10.1038/30918 PMID: 9623998

57. Rossi RA, Ahmed NK. The Network Data Repository with Interactive Graph Analytics and Visualization. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015; 29(1): 4292–4293. Available: https://ojs.aaai.org/index.php/AAAI/article/view/9277

58. Gregory S. Finding overlapping communities using disjoint community detection algorithms. Complex networks. 2009; 207: 47–61. https://doi.org/10.1007/978-3-642-01206-8_5

59. Newman MEJ. Assortative Mixing in Networks. Physical Review Letters. 2002; 89(20): 208701. https://doi.org/10.1103/PhysRevLett.89.208701 PMID: 12443515

60. Yang F, Li X, Xu Y, Liu X, Wang J, Zhang Y, et al. Ranking the spreading influence of nodes in complex networks: An extended weighted degree centrality based on a remaining minimum degree decomposition. Physics Letters A. 2018; 382(34): 2361–2371. https://doi.org/10.1016/j.physleta.2018.05.032

61. Gajewski Ł, Paluch R, Suchecki K, Sulik A, Szymanski B, Hołyst J. Comparison of observer based methods for source localisation in complex networks. Scientific Reports. 2022; 12: 5079. https://doi.org/10.1038/s41598-022-09031-0 PMID: 35332184

62. Zhang X, Zhang Y, Lv T, Yin Y. Identification of efficient observers for locating spreading source in complex networks. Physica, A. Statistical mechanics and its applications. 2016; 442: 100–109. https://doi.org/10.1016/j.physa.2015.09.017