IMPROVED VOTERANK ALGORITHM TO IDENTIFY CRUCIAL SPREADERS IN SOCIAL NETWORKS

Yaxiong Li, Xinzhi Yang

Xi'an Research Institute of High Technology, China

Received 16 March 2022, accepted 23 June 2022, published online 27 July 2022

In the field of complex networks, Identifying crucial spreaders with high propagation ability is an important aspect of research, especially in the background of the global spread of COVID-19. In view of this, a large number of ranking algorithms and their improved versions have been proposed to evaluate the importance of nodes in the network, such as degree centrality, betweenness centrality, and k-core centrality. However, most of these methods neglect to consider the average shortest path between important nodes in the process of node importance evaluation, which will be difficult to ensure that the initial crucial spreaders have a large influence on the network. Recently, the VoteRank algorithm proposed a new idea for identifying widely distributed key spreaders based on the voting mechanism, but there are some aspects of this algorithm that require improvement. In this paper, we propose a VoteRank improved by degree centrality, k-core, and h-index (DKHVoteRank) for identifying critical spreaders in the complex networks. We introduce additional metrics to optimize the voting mechanism of the VoteRank to ensure that our algorithm can identify a widely distributed spreaders with high importance in the network. We conducted simulation experiments based on the Susceptible–Infected–Recovered (SIR) model on 12 different complex network datasets, and the results show that our proposed algorithm performs significantly better than other benchmark algorithms in terms of propagation capability, propagation scale, and applicability of the algorithm.

DOI:10.5506/APhysPolB.53.8-A4

1. Introduction

Many activities of human society can be abstracted as networks for analysis [1], such as social networks [2], trade networks [3], transportation networks [4], power networks [5], *etc.* As society continues to evolve, these network systems are becoming increasingly complex and precise. Biomolecular networks [6] explore the functions of molecules in terms of their network structure. Online trade networks [7], through community segmentation,

identify people with the same preferences, so that products and advertisements can be precisely delivered to the groups in need. In the background of the global spread of the epidemic, the analysis of social networks and the identification of nodes in the network that have a greater capacity to spread play an important role in preventing the development of the spread of the epidemic [2, 8, 9].

Identifying the influential key nodes in the network has become a hot issue in the research of complex networks in recent years, which can be defined as the influence maximization (IM) problem [10]. On the one hand, identifying the influential key nodes in the network can effectively control the network, and in power networks, the robustness and anti-destructiveness of the network can be effectively enhanced through the maintenance of key nodes in the network [5, 11]. In social networks, the use of cutting off key nodes in the network can be effective in limiting the spread of, for example, rumors or epidemics [9, 12]. On the other hand, the identification of critical nodes can help us better understand the function and structure of the network, thus bringing more convenience to our life [13]. The research on identifying critical nodes in complex networks has also attracted the attention of many scholars and many ranking methods for evaluating the importance of networks have been proposed.

Traditional methods based on node centrality, such as degree centrality (DC) [14], betweenness centrality (BC) [15], and closeness centrality (CC) [16] are proposed to evaluate the importance of nodes in complex networks. DC reflects the importance of a node by the number of its neighbors in a complex network, while BC and CC are based on the global information of the nodes in the network and can achieve more accurate results than DC. However, these two methods need to calculate the shortest distance of node pairs, which is more complex and difficult to apply to large-scale network structures. Based on this, many scholars have also proposed some new methods.

Chen *et al.* [17] proposed a semi-local method for node importance evaluation by taking the degree information of neighboring nodes into consideration. Liu *et al.* [18] defined the importance of edges in the network based on the degree information of nodes and then proposed a new evaluation method — degree and importance of the line (DIL) method. Ren *et al.* [19] combined degree and clustering coefficient information to evaluate node importance. Some new centrality evaluation metrics have also been proposed in recent years to calculate the importance of nodes. Kitsak *et al.* [20] proposed a k-core centrality (KC) approach based on the location of the nodes in the network, arguing that the nodes at the center of the network are more important. Liu *et al.* [21] further distinguished the importance of nodes located at the same layer in the KC method by calculating the average

distance between the nodes and the node at the most central location of the network. Wang *et al.* [22] then used the information entropy of nodes to distinguish the importance of nodes located at the same layer. Yeruva *et al.* [23] proposed the Pareto-shell decomposition method using a Pareto front function based on the KC algorithm. Bea *et al.* [24] proposed the Neighborhood Coreness (NC) and Extended Neighborhood Coreness (ENC) methods by taking the k-core values of neighbor nodes and secondary neighbor nodes into consideration. Hirsch *et al.* [25] proposed a new method for evaluating the importance of nodes called the h-index, which was initially used to evaluate the influence of scientists and later was also widely used to evaluate the importance of nodes in complex networks. Based on the h-index method, Liu *et al.* [26] then proposed the Local h-index (LH-index) method, in which the h-index values of neighboring nodes are also used in the calculation of the node importance, making the evaluation results more accurate. Tong *et al.* [31] redefined the entropic centrality model of nodes in complex networks as a measure of the centrality of nodes. Sheikhahmadi *et al.* [32] proposed a Mixed Core, Degree, and Entropy (MCDE) method that uses KC, DC, and entropy for a comprehensive evaluation of the node importance.

In the above method, the node importance is measured by defining the node importance index. However, it does not guarantee that the nodes with high importance are widely distributed in the network. In the field of complex networks, a node with high importance means that it has more influence on its surrounding nodes and has a more important position in the global network [33]. When dealing with the IM problem, it is often necessary to identify a set of nodes with high propagation ability in the network. In order to better exert influence on the network, we want the identified nodes to be as widely distributed as possible in the network [10]. In a network, the distance between initial spreaders has a large impact on the propagation of information [34]. Some scholars proposed PageRank [27, 28], LeaderRank [29, 30], and other ranking algorithms based on random walks. Zhang *et al.* [35] proposed a voting mechanism-based method, called VoteRank, which can effectively identify important nodes widely distributed in the network. Sun *et al.* [36] applied the VoteRank algorithm to weighted networks and proposed an extended algorithm called WVoteRank. Kumar *et al.* [37] used the NC method to improve VoteRank and proposed the NCVoteRank algorithm. Guo *et al.* [38] proposed the EnRenew algorithm to use the node information entropy for the improvement of the VoteRank algorithm.

The VoteRank algorithm provides a new perspective for solving the IM problem. However, the algorithm has some drawbacks. First, the VoteRank algorithm treats the initial voting ability of all nodes as the same, without differentiating them according to their attributes, and the model design is not precise enough. Second, the algorithm only considers the attribute val-

ues of the nodes' neighbors when calculating the node scores and does not consider the influence of the nodes' attributes on the node scores. In addition, the VoteRank algorithm only weakens the voting ability of the nearest node to the selected node in the updating process, which does not ensure that the identified set of spreaders is distributed widely enough in the network. Kumar *et al.* [37] use the node NC value to address the above shortcomings and optimize the node score calculation process. However, the index is not accurate enough in reflecting the importance of nodes. Guo *et al.* [38] redefines the initial voting ability of nodes by information entropy, but do not optimize the calculation process of node scores.

Inspired by the VoteRank algorithm, to address the shortcomings of the algorithm, we have improved the algorithm in the following aspects:

(i) The initial voting ability of nodes is redefined using node degree values, and it is considered that nodes with larger degree values need to vote for more nodes and, therefore, need to have the stronger voting ability.

(ii) The node importance value is derived by combining the node degree value, the k-core value, and the h-index value, which measure the node capability from different aspects, and introduce the value into the calculation of the node voting score. Thus, the final score of the node reflects not only the attributes of the neighboring nodes, but also the importance of the node itself in the network.

(iii) The update phase of the VoteRank algorithm is improved by expanding the weakening range of nodes around the node selected in each round of voting and adjusting the weakening parameters considering the distance between nodes.

In this paper, we compare the DKHVoteRank algorithm with six other benchmark algorithms, and evaluate the initial spreaders identified by different algorithms using the SIR model [40] based on propagation dynamics theory [39]. The experimental results show that our proposed algorithm has better performance compared with other algorithms.

The framework of this paper is as follows: Section 2 introduces the related methods of our algorithm and other improved algorithms, and the details of the principles and steps of our proposed algorithm are clarified. Section 3 introduces the evaluation model, evaluation metrics, and data sets used in the experimental part. Section 4 analyzes the experimental results. Section 5 concludes the content of this paper.

2. Methods

We use $G(V, E)$ to denote an undirected unweighted complex network, $N = |V|$ to denote the set of nodes of the network, and $M = |E|$ to denote the set of edges of the network, where $V = \{v_1, v_2, \cdots, v_N\}$, $E = \{e_1, e_2, \cdots, e_M\}$.

2.1. DC

DC [14] is the most classical node ranking method which considers a node with more number of neighbors as more important. DC can reflect the importance of the node to some extent, but it is difficult to distinguish the nodes with the same degree. For example, the degree values of nodes 1 and 16 in Fig. 1 are both 8, so it is difficult to evaluate which of these two nodes is more important. Meanwhile, nodes with higher importance in the network do not always have higher degree values; for example, node 4 has a smaller degree value than node 16, but node 4 is closer to the center of the network than node 16. Therefore, relying on the DC method alone to identify key nodes in the network is not accurate enough. Chen *et al.* [17] realized the inadequacy of DC in measuring the importance of nodes and proposed a semi-local method (SL), which considers the information of node neighbors and secondary neighbors. This method effectively balances the problems of low relevance of the DC method and the complexity of global metrics calculation. The local importance $C_{\mathrm{L}}(v)$ of node v can be defined as

$$Q(u) = \sum_{w \in \Gamma(u)} N(w)\,, \tag{1}$$

$$C_{\mathrm{L}}(v) = \sum_{u \in \Gamma(v)} Q(u)\,, \tag{2}$$

where $\Gamma(v)$ denotes the set of neighbor nodes of node v, and $N(w)$ denotes the number of neighbor nodes and secondary neighbor nodes of node w.

2.2. KC

KC (also known as the k-shell method) is a node importance ranking method proposed by Kitsak *et al.* [20]. This method considers that the importance of a node depends on the location of the node in the network, and the nodes located at the center of the network have greater importance. The KC method lays the nodes as follows: first, all nodes in the network with degree 1 are removed, which will cause the degree value of the remaining nodes in the network to decrease, and further nodes in the network with degree 1 are removed until all nodes in the network have a degree value

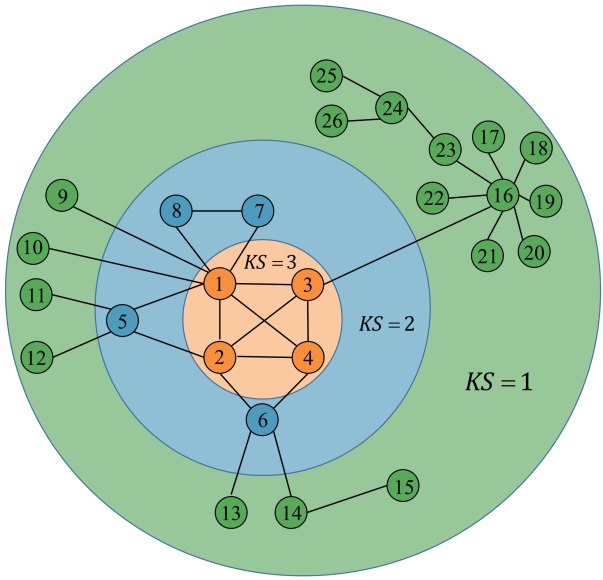


Fig. 1. An example network to explain the principles and steps of different algorithms, which is also used in [20, 24].

greater than 1. The KC value of the nodes removed in this step is defined as 1. After that, all the nodes in the network with degree 2 are removed from the network according to the same way until the degree of the remaining nodes is greater than 2. The nodes removed in this step are defined with the KC value of 2. After removing all the nodes in the network according to this method, the KC value of each node is obtained. The higher the KC value of the node, the closer the node is to the center of the network. In Fig. 1, the network nodes are divided into three layers according to the KC method, and the orange area indicates the nodes located at the center of the network. The KC method can effectively identify the nodes located at the center of the network, but the method does not distinguish well all nodes, for example, there are 26 nodes in Fig. 1, which are only divided into three layers, and the nodes in the same layer cannot be compared. Moreover, the nodes with the largest KC values in the network are generally concentrated (also obvious from Fig. 1), and if the node with the largest KC value is selected as the initial spreaders, it will lead to overlapping influence, which needs to be avoided as much as possible.

NC proposed by Bae and Kim [24] is based on the KC method in considering the KC value of a node and its neighbors to evaluate the importance of a node. The NC value of a node can be expressed as

$$\mathrm{NC}(v) = \sum_{w \in \Gamma(v)} ks(w)\,, \tag{3}$$

where $\Gamma(v)$ denotes the set of neighbors of node v, and $ks(w)$ denotes the KC value of node w. Based on the NC method, the ENC method is further proposed, and the ENC values of the nodes can be expressed as

$$\mathrm{ENC}(v) = \sum_{w \in \Gamma(v)} \mathrm{NC}(w)\,, \tag{4}$$

where $\mathrm{NC}(w)$ denotes the NC value of node w.

2.3. h-index

h-index [25] is a metric proposed by Hirsch to measure the research contribution of scientists. This method is defined as the h-index value of a scientist if he has N papers, of which h papers are cited more than h and the remaining $(N - p)$ papers are all cited less than h. The h-index value of this scientist is h. In recent years, this method has also been used to evaluate the importance of nodes in complex networks due to the rationality of the evaluation. When this method is used to evaluate nodes in complex networks, then it can be defined as when a node has h neighbors whose degree values are all greater than h and the remaining neighbors whose degree values are all less than h, the node has an h-index value h. Take node 1 in Fig. 1 as an example, the degree values of the neighboring nodes of node 1 are 5, 4, 4, 4, 2, 2, 1, 1. Node 1 has 4 neighbors with degree values greater than 4, and the degree values of the remaining nodes are less than 4, so the h-index value of node 1 is 4. The h-index method measures the importance of a node by the number of high-quality neighbor nodes of that node. It does not consider the other neighbors of the node, which leads to the conclusion that the method is not sensitive to some changes in the network, for example, if the links between node 1 and four nodes 17, 18, 19, 20 are added, the degree of node 1 increases by 4, but its h-index value does not change.

Liu *et al.* [26] proposed the LH-index method based on the h-index method, which takes the h-index values of the neighbors into consideration, and the LH-index value of node i can be defined as

$$\mathrm{LH}_{\mathrm{index}}(i) = h_{\mathrm{index}}(i) + \sum_{v \in \Gamma(i)} h_{\mathrm{index}}(v)\,. \tag{5}$$

The LH-index method can better distinguish node importance compared to the h-index method, and the h-index values of nodes are affected by neighboring nodes and, therefore, are more sensitive to changes in the network.

2.4. VoteRank

Zhang *et al.* [35] proposed the VoteRank algorithm, based on a voting mechanism, to select the most influential nodes based on the scores of the nodes in each round of voting. Each node in the network contains two attributes, $\{S_v, V_{a_v}\}$, where S_v is used to record the voting score of node v after each iteration, and V_{a_v} indicates the voting ability of node v during each iteration. The voting score of a node is equal to the sum of its neighbors' voting ability and the VoteRank algorithm goes through the following five steps:

Step 1: Initialize. Initialize the voting score S_v and voting ability V_{a_v} of all nodes in the network to 0 and 1.

Step 2: Vote. In this phase, each node votes on its neighbors, and each receives all votes from its neighbors. The voting score of node v in the T^{th} round of voting $S_v(T)$ can be expressed as

$$S_v(T) = S_v(T-1) + \sum_{i \in \Gamma(v)} V_{a_i}(T-1)\,. \tag{6}$$

Step 3: Select. The node with the highest voting score is selected based on the results of the current round of voting. The selected node $v_{T\,\max}$ will not participate in the next round of voting, thus changing the voting ability of this node to 0.

Step 4: Update. In order to make the selected nodes as diffuse as possible, the voting ability of the selected node's neighbors needs to be diminished. The diminished node voting ability can be defined as

$$V_{a_v} = \begin{cases} V_{a_v} - \delta & \text{if} \quad V_{a_v} - \delta > 0 \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

where $\delta = \langle k \rangle / \left(\langle k^2 \rangle - \langle k \rangle \right)$ denotes the reduction coefficient of the node voting ability, and $\langle k \rangle$ denotes the average degree of the network.

Step 5: Repeat. Repeat the process from Steps 2 to 4 until the top k nodes are selected out.

Kumar *et al.* [37] argue that the voting ability of nodes needs to be differentiated according to the topological position of nodes in the network, and the NCVoteRank algorithm is proposed to improve the voting ability of nodes by introducing the NC value of nodes, and the node NC value is calculated according to Eq. (3). The NCVoteRank algorithm also first initializes the voting ability V_{a_v} and the scores S_v of all nodes in the network to 1 and 0. The following formula is used to calculate the node scores in the voting phase:

$$S_v = \sum_{i \in \Gamma(v)} \left(V_{a_i} \times \mathrm{NC}(i) \times (1-\theta) + V_{a_i} \times \theta \right), \tag{8}$$

where NC(i) is the node's neighborhood coreness value, θ is an adjustable parameter that takes values in the range of [0, 1], which is used to adjust the weight of the node's NC value, after which the node with the highest score in this round of voting is identified and the information about the node and the node's neighboring nodes is updated. After that, the next voting is performed until the first n initial propagation nodes are selected.

Inspired by the VoteRank algorithm, Guo *et al.* [38] proposed the EnRenew algorithm by using the node information entropy as the initial voting ability of a node. The information entropy E_v of node v can be calculated by

$$E_v = \sum_{u \in \Gamma(v)} H_{uv} = \sum_{u \in \Gamma(v)} -P_{uv} \log P_{uv} \,, \tag{9}$$

where $p_{uv} = \frac{d_u}{\sum_{l \in \Gamma(v)} d_l}$, and H_{uv} denotes the propagation ability that node v receives from node u. The EnRenew algorithm selects the node in the network with the largest propagation ability as the selected node, and weakens the propagation ability of the node's l-length reachable nodes. The weakened propagation ability can be calculated by

$$H_{u^{l-1}u^l} = H_{u^{l-1}u^l} - \frac{1}{2^{l-1}} \frac{H_{u^{l-1}u^l}}{E_{\langle k \rangle}} \,, \tag{10}$$

where u^l indicates that the distance between node u^l and the selected node is l, and $E_{\langle k \rangle} = -\langle k \rangle \frac{\langle k \rangle}{n} \log \frac{\langle k \rangle}{n}$ is the information entropy of any node in the k-regular graph network.

2.5. Our method

The previous sections introduced several traditional methods for identifying important nodes in the complex networks and their respective improved algorithms. The DC method reflects the importance of a node by the number of neighboring nodes, and the larger degree value means it has more influence in the network locally; the KC method focuses on the location of a node in the network, and the larger the k-core value, the closer the node is to the center of the network. The h-index method is more concerned with the number of high-quality neighbors around the node. The above three algorithms have low computational complexity and can better measure the node importance from different dimensions.

The VoteRank algorithm selects influential spreaders in the complex networks by using a voting mechanism. We believe that a more suitable node-importance metric should be used to improve the VoteRank algorithm — one that better reflects the position of the nodes in the network topology during the voting process. Therefore, we propose an algorithm called DKHVoteRank

and we use three methods DC, KC, and h-index to improve the VoteRank algorithm. The details of DKHVoteRank algorithm are as follows:

(i) First, the degree value d_i, KC value k_i, and h-index value h_i of all nodes in the complex network are calculated. The above three metrics are combined using the homotopy function to obtain the local importance of the nodes p_i, which is calculated as follows:

$$p_i = \frac{d_i}{\sqrt{\sum_{j=1}^{N} d_j^2}} + \frac{k_i}{\sqrt{\sum_{j=1}^{N} k_j^2}} + \frac{h_i}{\sqrt{\sum_{j=1}^{N} h_j^2}}, \tag{11}$$

where N denotes the number of nodes in the network. The degree value of a node can reflect the local influence ability of the node in the network to some extent, the k-core value reflects the importance of the node's position in the network, and the h-index value reflects the importance of the node through the number of high-quality nodes in the node's neighborhood. We want to obtain a metric that can synthesize the importance of different aspects of the nodes, so we use the homotopy function $u(x) = x/\sqrt{\sum x^2}$ [19, 41] to process d_i, k_i, and h_i simultaneously, so that it can correctly reflect the combined results of different forces.

(ii) Initialize node score and voting ability. In this phase, the node voting score is initialized to 0, and the voting ability V_{a_v} is calculated according to the following formula:

$$V_{a_v} = \log\left(e + \frac{k_v}{k_{\max}}\right), \tag{12}$$

where e is a constant that represents the base of the natural logarithmic function, $k_{\max}$ denotes the maximum value of the node degree in the network. In the VoteRank algorithm, all nodes' initial voting ability in the network is set to 1 in the initialization phase. We believe that the initial voting ability of nodes should be differentiated according to the degree value of nodes. The larger the degree value of nodes which means that nodes have more neighboring nodes and need to cast more votes in the voting phase, the stronger the voting ability of nodes themselves should be. Therefore, the logarithmic function is used in this step to describe the trend of node voting ability with the degree value.

(iii) Voting phase. In the voting phase, each node in the network receives votes from its neighbors and votes for its neighbors. In calculating the node score for each round of voting, the node score is calculated by multiplying the node's local importance p_i with the node's voting ability V_{a_v}, calculated as follows:

$$S_v(i) = \sum_{i \in \Gamma(v)} (V_{a_v} p_i) \ . \tag{13}$$

The VoteRank algorithm calculates the final score of a node by summing up the voting ability of the node's neighboring nodes. It calculates the final score of a node only by the attribute values of the neighboring nodes, without considering the influence of the node's own attributes on the node's score, so at this stage, we take the node's importance value p_i into account in the calculation of the node's score. After calculating the scores of all nodes, the node with the highest score is selected in this round of voting, and the selected node will not participate in the subsequent voting process.

(iv) Update node attribute values. The node with the top voting score in this round is selected, and its voting ability is set to 0. We assume that node v_T is the selected node for the T^{th} round of voting. Then, we update the voting ability values of the nearest and the next-nearest neighbors of node v_T as follows:

$$Va_v = \begin{cases} V_{a_v} - \delta & \text{if} \quad V_{a_v} - \delta > 0 \\ 0 & \text{otherwise} \end{cases} \ , \tag{14}$$

where $\delta = \frac{1}{\langle k \rangle \times d(v_k, v_T)}$ denotes the reduction coefficient of the voting ability, and $d(v_k, v_T)$ denotes the distance between v_k and v_T. This step makes it more difficult to elect nodes in the domain of the selected nodes in the voting process thereafter by weakening the voting ability of the neighbors of the selected nodes, so that the identified set of selected nodes can be widely distributed in the network. At the same time, we weaken the voting ability of nodes within distance 2 from the selected node. Considering the negative correlation between the influence and the distance between nodes, the node distance is taken into account in the weakening mechanism, and the farther the distance from the selected node, the lower the weakening value is set.

(v) Iteration phase. Repeat Steps *(iii)* to *(iv)* until the top k nodes are selected.

The detailed steps of the DKHVoteRank algorithm are shown in Algorithm 1. In lines 2–4, the local importance of each node in the network is calculated according to Eq. (11), and we denote the number of nodes and edges in the network by n, m, respectively, and the computational complexity of this step is $O(n)$. In lines 6–19, the voting phase of the algorithm is entered. First, the nodes' scores are calculated after each round of voting, and the node with the highest score is selected after which the neighbors and secondary neighbor information of the selected node are updated, so the computational complexity is $O(n\langle k\rangle^2)$, where $\langle k\rangle$ represents the average degree of the network, $\langle k\rangle = \frac{2m}{n}$. The above steps need to be repeated s times, and s denotes the number of initial spreaders. Ultimately, the computational complexity of the algorithm can be expressed as $O(n + sn\langle k\rangle^2)$, which can also be approximated as $O(n\langle k\rangle^2)$, since the value of s is generally much smaller than n.

Algorithm 1 DKHVoteRank

Input: a complex network $G(V, E)$ with $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_n\}$, number of initial spreaders *topk*

Output: S includings *topk* nodes

1: $S = \varnothing$
2: **for all** $v \in V$ **do**
3: $p_i = \frac{d_i}{\sqrt{\sum_{j=1}^{N} d_j^2}} + \frac{k_i}{\sqrt{\sum_{j=1}^{N} k_j^2}} + \frac{h_i}{\sqrt{\sum_{j=1}^{N} h_j^2}}$ ▷ compute the node local importance
4: $Va_v = \log\left(e + \frac{k_v}{k_{\max}}\right)$ ▷ initialization the voting ability
5: **end for**
6: **while** $|S| \leq topk$ **do**
7: **for all** $v \in V$ **do**
8: $S_v(i) = \sum_{i \in \Gamma(v)} (Va_v * p_i)$
9: **end for**
10: Add v_i to S, delete v_i form V, where $v_i = \arg\max_v \{S_v\}$
11: **for** $v_j \in \Gamma(v_i)$ **do** ▷ Eliminating nodes, which $v_i's$ reachable in two hops
12: $Va_j = Va_j - 1/\langle k\rangle$
13: $S_j = S_j - Va_i$
14: **for** $v_k \in \Gamma(v_j)$ **do**
15: $Va_k = Va_k - 1/(2 * \langle k\rangle)$
16: $S_k = S_k - Va_j$
17: **end for**
18: **end for**
19: **end while**
20: **return** S;

3. Experimental setup

There are two methods to evaluate the importance of the initial spreaders determined by different algorithms [42]. The first approach uses the node deletion method, which considers the importance of a node as equivalent

to the impact on the network after the deletion of that node. The second approach is based on a propagation dynamics model, where the identified initial spreader is used as a source of information propagation and the importance of the initial spreaders is evaluated by simulating the propagation in the network. The first method, which needs to calculate the distance of all node pairs in the network when calculating the network operation efficiency, has a large computational complexity and is difficult to be applied to large-scale network structures. In the propagation dynamics model, by simulating the process of information propagation in the network, the calculation results are more straightforward, and this method has also become the major method for evaluating the initial spreaders at present.

3.1. Spreading model

The SIR model [43, 44] has been known as the most commonly used propagation model due to its good operability and applicability. In the SIR model, the nodes in the network are classified into three categories, respectively, susceptible nodes (S), infected nodes (I), and recovered nodes (R). In the beginning, a small number of nodes in the network are selected as infected nodes, which are in state I, and the remaining nodes in the network are defined as susceptible nodes in state S. In each step of propagation, the infected nodes have a certain probability to assimilate the susceptible nodes in their neighbors into infected nodes, and the infection probability is defined as β. At the same time, the infected nodes in the network have a certain probability of recovery and are transformed into recovered nodes, and the recovery probability of the nodes is defined as λ. The infected probability has a threshold value β_{th}, and when the infection probability is less than this threshold, the information cannot be effectively propagated in the network. Therefore, in order to make the propagation process more rapid so that we can observe the differences between different initial spreaders in the propagation process, we set the infection probability $\beta = 1.5\beta_{\text{th}}$, where $\beta_{\text{th}} = \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle}$ and $\langle k \rangle$ denotes the average of node degrees in the network. In the SIR model, the infection rate is defined as the ratio of the infection probability to the recovery probability, $\zeta = \frac{\beta}{\lambda}$, and this metric also has a significant impact on the process of information dissemination in the network.

3.2. Performance metrics

3.2.1. Network efficiency

Network efficiency η [19] is a common metric used to evaluate network connectivity. The higher the network efficiency, the stronger the connectivity

between nodes in the network. The formula for calculating network efficiency can be expressed as

$$\eta = \frac{1}{n(n-1)} \sum_{v_i \neq v_j} \frac{1}{d(v_i, v_j)}, \tag{15}$$

where n denotes the number of nodes in the network and $d(v_i, v_j)$ denotes the shortest path between nodes v_i and v_j. When a node is removed from the network, the edges connected to that node are also removed at the same time, which may cause the shortest path between some pairs of nodes in the network to be interrupted and the distance between node pairs to increase, thus leading to a decrease in network efficiency. Therefore, the percentage of the decrease in network efficiency after removing a node can be an important indicator of the node's importance in the network. Assuming that the network efficiency is η_0 before the nodes are removed and the network efficiency becomes η' after the nodes are removed, then after node v_i is removed, the rate of decrease in network efficiency μ_i can be expressed as

$$\mu_i = 1 - \frac{\eta'}{\eta_0}. \tag{16}$$

The network efficiency decline rate can be used to measure the importance of a single node in the network, while the importance of a certain set of nodes in the network can be evaluated.

3.2.2. Propagation scale

Under the SIR model, in each iteration step, infected nodes infect neighboring susceptible nodes to achieve propagation in the network. At the same time, infected nodes have a certain probability of becoming recovered nodes in the propagation process, so the number of infected nodes in the network will gradually increase with time and then decrease. When the number of infected nodes decreases to 0, only susceptible nodes and recovered nodes are left in the network and the network stops spreading. Based on this, we can use $F(t)$ to denote the ratio of infected nodes and recovered nodes to the total number of nodes, which is a curve that changes with time during network propagation. It can be used as an indicator to evaluate the propagation ability of initial spreaders. $F(t)$ can be expressed as

$$F(t) = \frac{n_{\mathrm{I}}(t) + n_{\mathrm{R}}(t)}{n}, \tag{17}$$

where $n_{\mathrm{I}}(t)$ and $n_{\mathrm{R}}(t)$ denote the number of infected nodes and recovered nodes in the network, respectively, at time t. When the number of infected

nodes drops to 0, $F(t)$ reaches its maximum value $F(t_c)$, which can be expressed as

$$F(t_c) = \frac{n_{\mathrm{R}}(t_c)}{n}, \tag{18}$$

where t_c indicates that the number of infected nodes drops to 0 at the moment t_c. This can be used as a metric to evaluate the propagation scale of the initial spreader.

3.2.3. Average distance between spreaders

The average distance is an important index to evaluate the dispersion of the initial spreader, which has an important impact on maximizing influence. With the limited number of initially selected nodes, we want the selected nodes to be as dispersed as possible in the network to improve the coverage area during propagation. In most real networks, the node distribution shows the phenomenon of community aggregation, and if the selected nodes are too concentrated, it is difficult to spread the information to other communities effectively. The average shortest path can be found by the distance between any two nodes in the node set, which is calculated as follows:

$$L_s = \frac{2\sum_{v_i \neq v_j \in S} D_{ij}}{s(s-1)}, \tag{19}$$

where S denotes the initial spreader selected by different algorithms, s denotes the number of nodes in S, and D_{ij} denotes the shortest distance between nodes v_i and v_j. Larger values of L_s indicate that the spreaders are more widely distributed and have better coverage in the network.

3.3. Data description

To test the performance of the algorithm, we performed operations using 12 real network datasets, selected with different data sizes and data sources. These datasets are frequently used in research on complex networks. The following is a description of the datasets used for the tests:

(1) karate: a small social network dataset containing interpersonal relationships and interconnections among 34 members of the Karate Club of America [45];

(2) dolphins: an undirected social network that portrays the interactions and community distribution of 62 dolphins [46];

(3) jazz: this dataset contains the interactions of a network of jazz musicians [47];

(4) CEnew: a biological metabolic network [48];

(5) email: a network of email exchanges among members of Rovira Virgili University [49];

(6) Netscience: a coauthorship network of scientists working on network theory and experiments [50];

(7) USAir: a network of the US air transportation system in 2010 [51];

(8) hamster: a friendship network between users of the website `hamsterster.com` [52];

(9) Facebook: a crowd-sourced dataset containing information about the social circles of Facebook users [53];

(10) power: a power grid network in the USA [54];

(11) router: reflects the Internet topology at the router level [55];

(12) condmat: a coauthorship network between researchers on the topic of condensed matter [56].

Some of their basic network properties are listed in Table 1.

Table 1. Basic characteristics of the 12 complex network datasets, where $\langle k \rangle$ denotes the average degree of the network, and denotes the threshold of infected probability in the SIR model.

Network	n	m	$\langle k \rangle$	β_{th}
karate	34	78	4.59	0.148
dolphins	62	78	5.13	0.172
jazz	198	2742	27.70	0.027
CEnew	453	2025	8.94	0.026
email	1133	5451	9.62	0.057
Netscience	1461	2742	3.75	0.168
USAir	1574	17215	21.87	0.009
hamster	2426	16631	13.71	0.024
Facebook	4039	88234	43.69	0.009
power	4941	6594	2.67	0.348
router	5022	6258	2.49	0.079
condmat	23133	93497	8.08	0.047

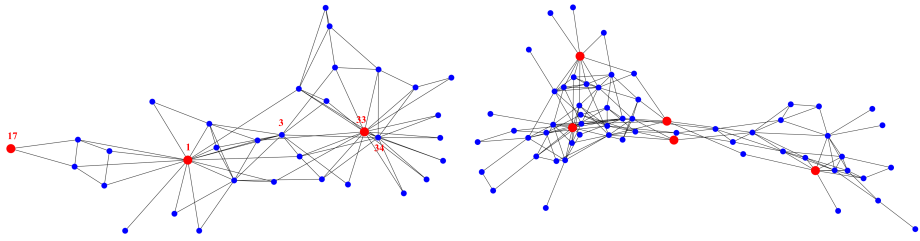
4. Experiment results

In Section 3, we have presented two methods based on the node deletion method and propagation dynamics simulation for evaluating the accuracy

of different identification algorithms, and the main metrics of both evaluation algorithms have been described. In this chapter, we set up experiments for evaluation. To better show the details of the algorithms, we first take two small-scale datasets, karate and dolphins, as examples to compare the identification results of different algorithms. Considering that the node deletion method is difficult to apply to large-scale network data sets and that the simulation results based on the propagation dynamics model are more convincing for evaluating IM problems, in the second part of this chapter, we evaluate different algorithms mainly based on the propagation dynamics model. We select the VoteRank algorithm and its two improved algorithms NCVoteRank, EnRenew, and the improved algorithms SL, ENC, and LH-index for DC, KC, and h-index, as benchmark algorithms for comparing our proposed algorithms.

4.1. Comparison in small networks

Figure 2 shows the identified nodes in the karate and dolphins networks by our algorithm, which are marked in red. Figure 3 shows the nodes identified by the other six algorithms in the above two networks, (a)–(f) in Fig. 3 show the identification results of the different algorithms in the karate network, and (g)–(l) show the crucial nodes identified in the dolphins network. Table 2 shows the number of the set of nodes identified by different algorithms, ratio of nearest neighbors (RNN, which can be used to reflect the influence of the set of nodes on the network) of the selected nodes to the number of network nodes, and the decrease rate of the network efficiency after removing the selected set of nodes.



(a) karate (b) dolphins

Fig. 2. The karate and dolphins networks, the nodes marked in red are a set of influential nodes identified by the DKHVoteRank algorithm (For purposes of description, the number of some nodes in the karate network is labeled).

In the karate network, the node numbers identified by our algorithm are 33, 1, 17, and these three nodes have a large distribution range in the network — the number of nearest neighbor nodes of the three nodes accounts for 94.1% of the total number of nodes in the network, which can directly

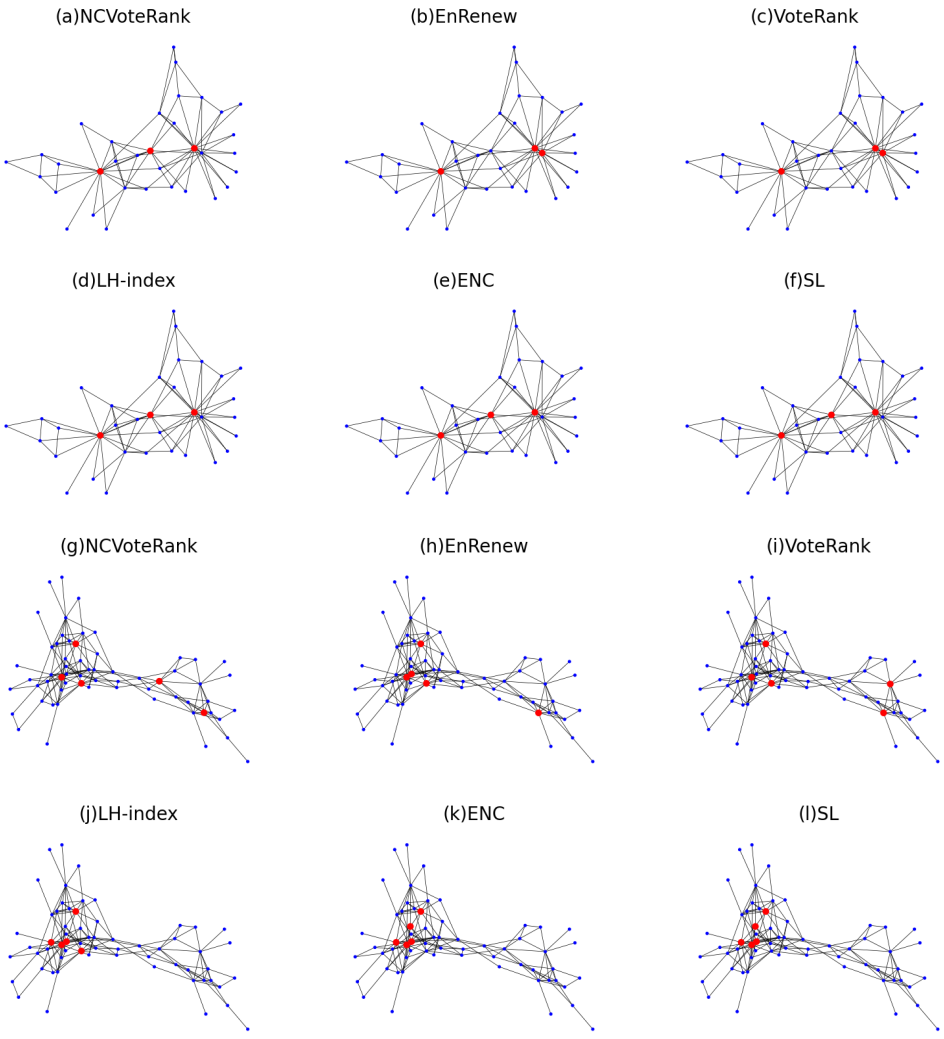


Fig. 3. The identification results of the six benchmark algorithms in the karate and dolphins networks, the nodes marked in red indicate the nodes identified by the algorithms as influential, where (a)–(f) indicate the identification results of the different algorithms in the karate network, respectively, and (g)–(l) indicate the identification results in the dolphins network.

influence the majority of nodes in the network and rank first among the seven algorithms. Nodes 33 and 1 identified by our algorithm are also identified by the other six algorithms, which also shows that these two nodes have a very important role in the karate network. For small-scale networks, the results identified by different algorithms may appear to be highly similar.

Table 2. The identification results of the seven algorithms for the karate and dolphins networks are presented in the table with the numbers of the selected node sets, the ratio of the nearest neighbors (RNN) of the selected node sets to the total number of network nodes, and the decrease rate of the network efficiency (μ) after removing the selected node sets.

Method	karate			dolphins		
	Selected nodes	RNN	μ	Selected nodes	RNN	μ
DCHVoteRank	[34, 1, 17]	**0.941**	0.442	[14, 57, 28, 51, 36]	0.645	**0.306**
NCVoteRank	[3, 34, 1]	0.912	0.556	[14, 45, 13, 1, 20]	0.645	0.228
EnRenew	[1, 34, 33]	0.912	**0.658**	[14, 45, 37, 20, 57]	0.613	0.258
VoteRank	[34, 1, 33]	0.912	**0.658**	[14, 45, 17, 20, 57]	**0.677**	0.275
LH-index	[1, 34, 3]	0.912	0.556	[14, 45, 37, 33, 20]	0.467	0.274
ENC	[1, 34, 3]	0.912	0.556	[14, 37, 45, 33, 50]	0.435	0.243
SL	[1, 3, 34]	0.912	0.556	[14, 37, 45, 33, 50]	0.435	0.243

Both EnRenew and VoteRank algorithms consider the node 34 to be very critical for the network. We can clearly see from Fig. 2 that nodes 33 and 34 of the network are located adjacent to each other and share most of their neighboring nodes, which means that these two nodes have a large overlap of influence on the local scope of the network. The simultaneous selection of these two nodes will have a greater impact on the efficiency of the network, but there may be a "$1 + 1 < 2$" situation in terms of exerting influence on the network, which needs to be avoided. The four algorithms NCVoteRank, LH-index, ENC, and SL identify node 3 in the network, which is located in the hub position of the network, however, it is directly connected with both nodes 1 and 33, which causes the farthest distance between the three identified nodes to be only 2. Meanwhile, node 3 shares all neighbors with the other two nodes, therefore, the selection of node 3 also generates a certain amount of influence duplication. Our chosen node 17, although located at the edge of the network, has less overlap with the influence range of the other two nodes. It has to be said that if only node 17 is compared with nodes 1 and 34, node 17 is much less important for the network, but the combination of node 17 with nodes 1 and 33 is seen to have a stronger influence on the network than the other algorithms, which is also evident from the subsequent experimental results of the simulated propagation. In the dolphins network, our algorithm identifies nodes distributed at key locations in different parts of the network, which have the greatest impact on the network efficiency, as well as a higher number of nearest neighbors than most of the compared algorithms. Overall, our algorithm pays more attention to the influence of the identified set of nodes on the network compared to the importance of individual nodes.

4.2. Experiment result based on SIR model

In Section 3, we have illustrated the importance of the infection rate for the initial spreaders in the network propagation process, and the related evaluation metrics. Meanwhile, the proportion of initial spreaders has a large impact on the final infection scale in the network. In this section, we design four sets of experiments to observe the performance of the spread of the initial spreaders identified by our proposed DKHVoteRank algorithm and the other six benchmark algorithms from different perspectives in the 12 datasets listed in Table 1.

4.2.1. The variation curve of propagation scale with time for different initial spreaders

In order to observe more intuitively the initial spreaders identified by different algorithms in the network propagation, in Fig. 4, we show the propagation scale curve with time during the propagation. The purpose of this experiment is to observe the propagation ability of the initial spreaders identified by different algorithms under the same conditions.

From the experimental results, we can see that our proposed DKHVoteRank algorithm has better performance compared to the other six algorithms. Specifically, among the 12 datasets, the spreader identified by DKHVoteRank algorithm has the strongest propagation ability, especially in jazz, CEnew, email, hamster, and condmat datasets. The spreader identified by our algorithm has a higher slope in the initial propagation stage and can reach stability at a faster speed, and the final propagation scale is significantly higher than other algorithms, which means the spreader identified by our algorithm has a faster propagation speed and stronger propagation ability. In addition, our algorithm has strong applicability and achieves better performance in different datasets, which is also significantly better than other algorithms, for example, in email, Netscience, and condmat datasets, EnRenew algorithm performs second only to our proposed algorithm, but in karate, dolphins, jazz, USAir, and router, the performance is not outstanding. Similarly, the VoteRank algorithm has the highest propagation speed in the initial stage of propagation in USAir and Facebook datasets, but also does not perform well in karate and email networks.

It is worth mentioning that we choose three improved methods based on h-index, KC, and DC for reference, LH-index, ENC, and SL, respectively, and it can be seen from almost all experimental results that the performance of the above three algorithms is significantly weaker than the other four algorithms based on the voting mechanism. The possible reason is that the above three algorithms only evaluate the value of individual nodes and neglect to consider node value from the global perspective of the network,

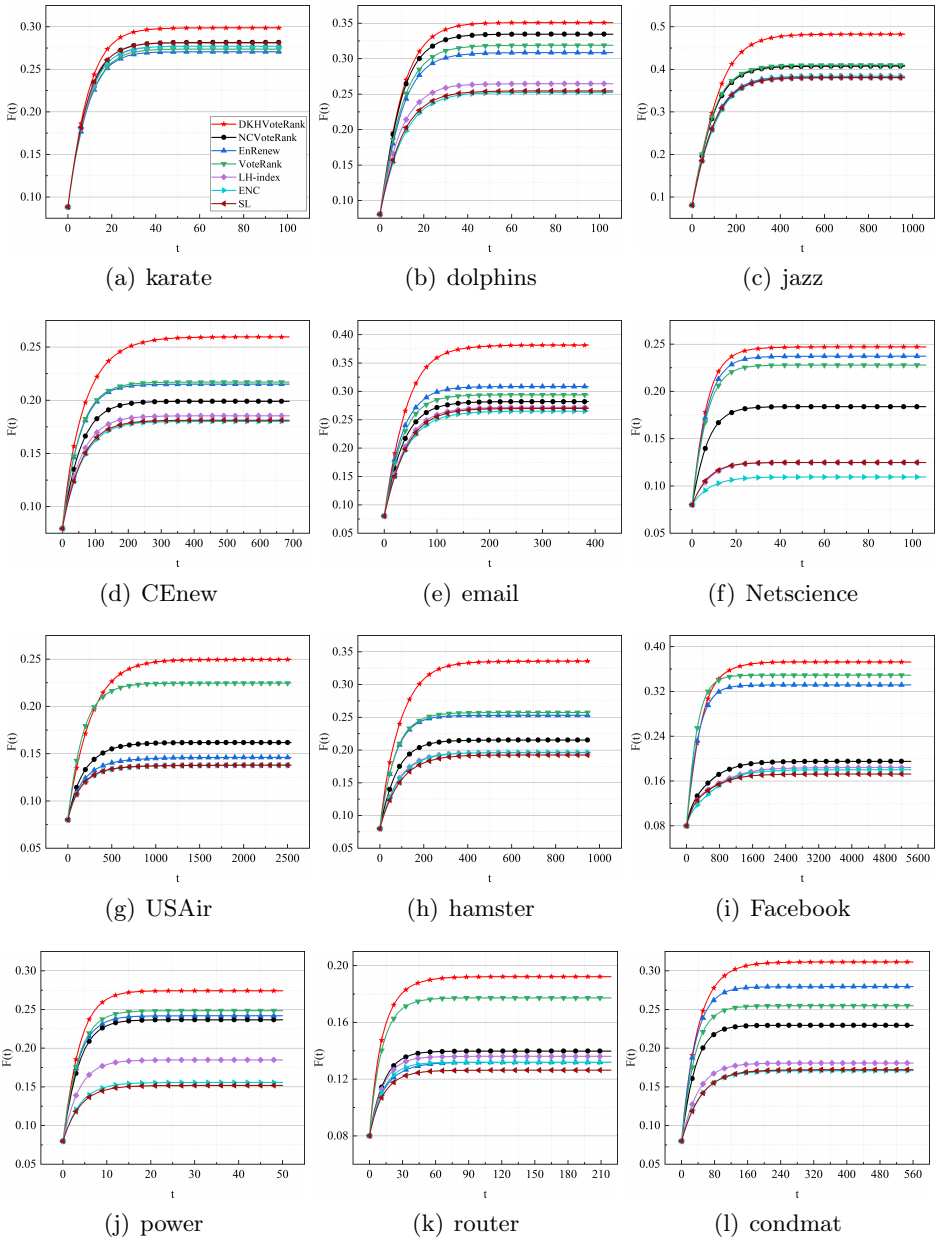
which makes it difficult to ensure that the initial spreader is widely distributed in the network, thus is makes the experimental results significantly weaker than the algorithms based on the voting mechanism. This view can also be verified in Fig. 7.

4.2.2. The variation of final propagation scale under different proportions of initial spreaders

This experiment compares the size of the initial node propagation ability identified by different algorithms by adjusting the proportion of initial spreaders. From Fig. 5, it can be seen that as the proportion of initial spreaders increases, the final infection scale in the network is constantly getting larger, and the differences between different algorithms become more and more obvious. From the experimental results of the karate, jazz, email, US-Air, and hamster datasets, the curves of infection scale with the proportion of initial spreaders of other benchmark algorithms roughly overlap, while the DKHVoteRank algorithm performs significantly better than the other algorithms, which indicates that our proposed algorithm can effectively identify the important nodes that are ignored by other algorithms. From the experimental results of 12 datasets, we can find that the DKHVoteRank algorithm can achieve better experimental results than other algorithms with different initial node ratios, which also shows the stability and applicability of the algorithm.

4.2.3. The change of final propagation scale at different infection rates

The lower the recovery probability, the more difficult it is for a node to transform from an infected node to recovered node, and the more persistent the impact on its neighboring nodes, and the larger the final propagation scale will be. From Fig. 6, it can be concluded that when the number of nodes in the network is small, different algorithms are not greatly affected by the infection rate, but when the network size gradually increases, different algorithms show differences. The experimental results also show that our proposed algorithm still maintains a better performance compared to other algorithms under the influence of different infection rates.



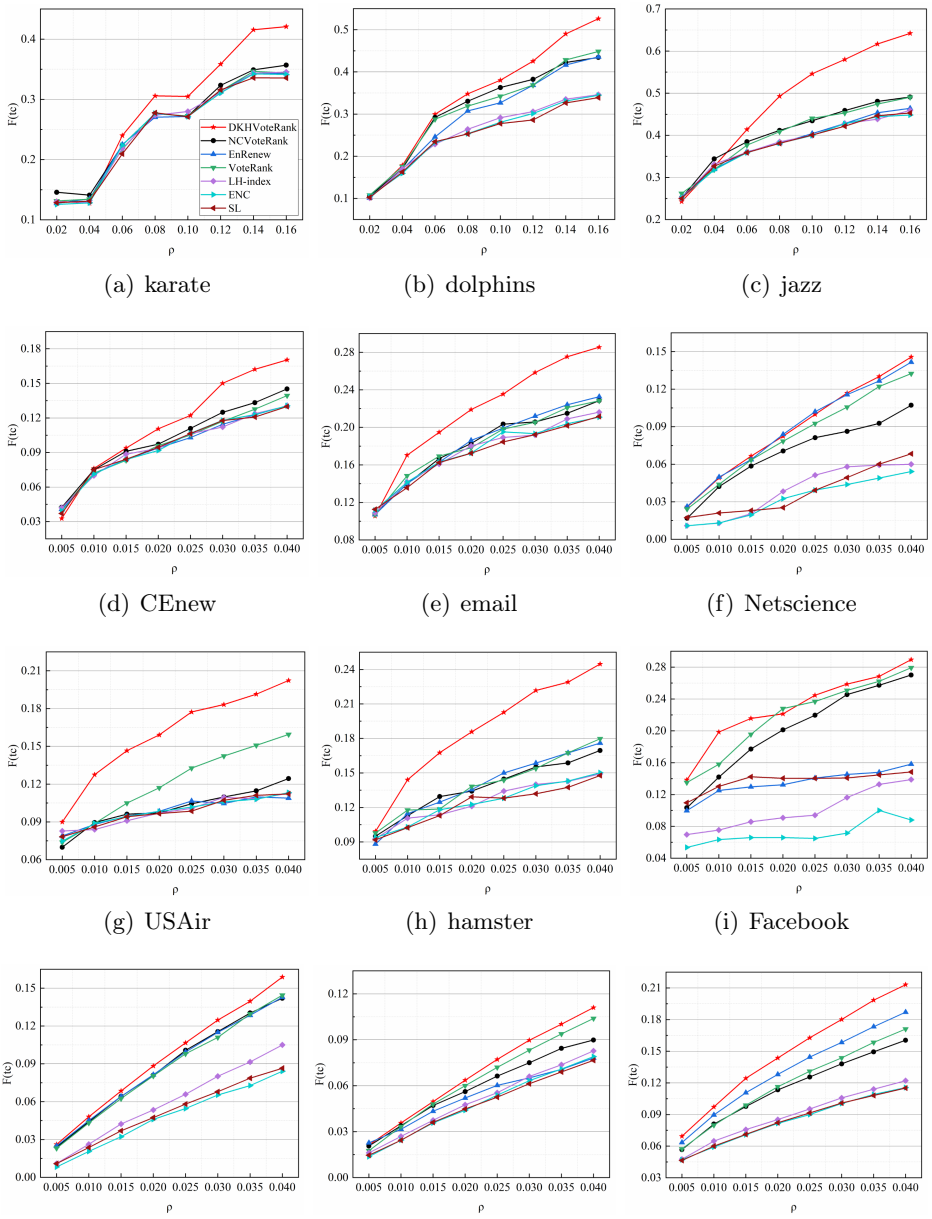
(a) karate (b) dolphins (c) jazz

(d) CEnew (e) email (f) Netscience

(g) USAir (h) hamster (i) Facebook

(j) power (k) router (l) condmat

Fig. 4. The change the curve of infection scale with time during the spread of the initial spreaders identified by different algorithms. The results are averaged over 1000 simulation experiments and we set the infection probability at $\beta = 1.5\beta_{\text{th}}$, initial spreaders ratio $\rho = 0.02$, and infection rate $\zeta = 1.25$, to ensure the smooth processes of the spread.



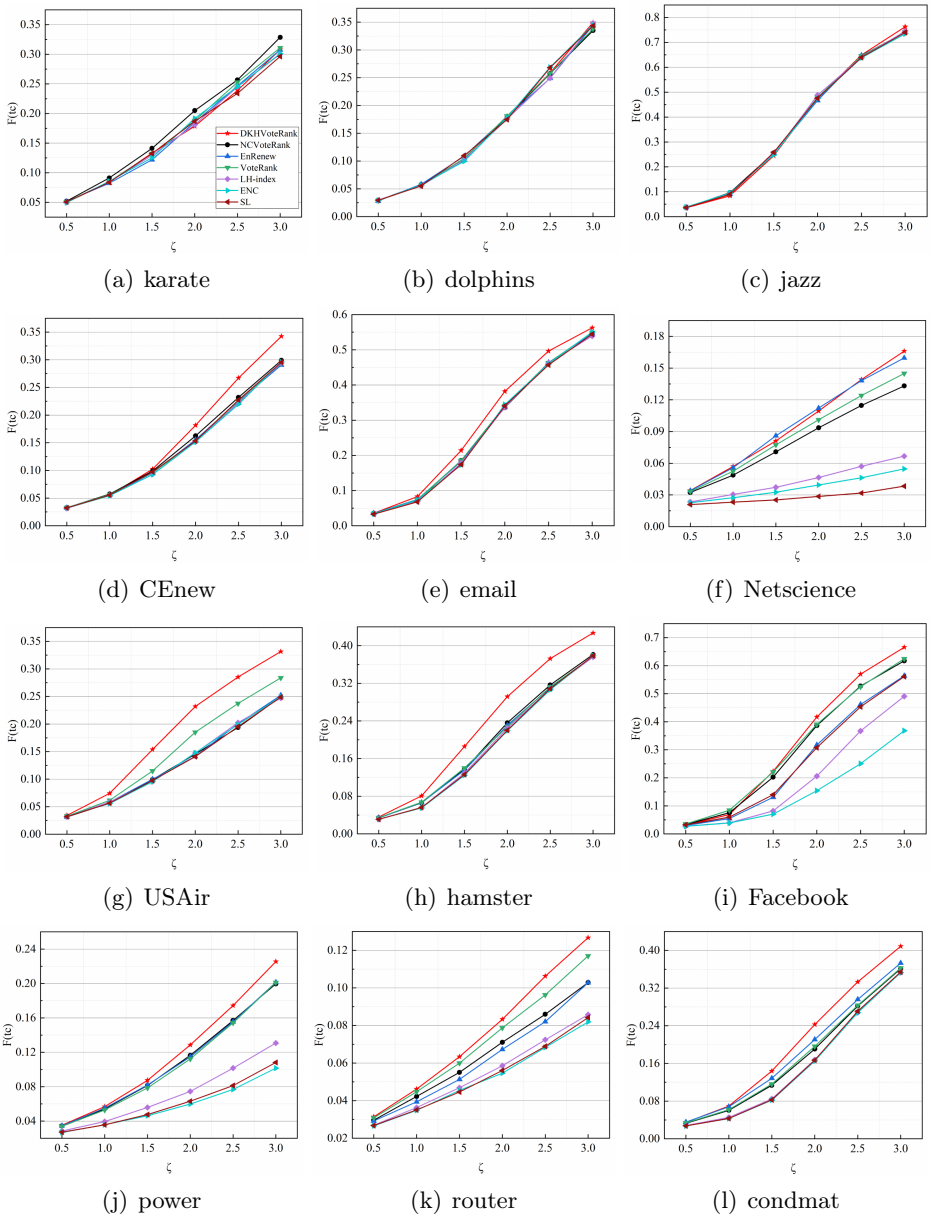
(a) karate (b) dolphins (c) jazz

(d) CEnew (e) email (f) Netscience

(g) USAir (h) hamster (i) Facebook

(j) power (k) router (l) condmat

Fig. 5. The change curve of the final infection size with the proportion of initially infected nodes, and the results are also selected as the average of 1000 simulations, and the infection proportion $\zeta = 1.25$, due to the lower number of nodes in the karate, dolphins, and jazz networks, the initial spreaders ratio is set at [0.02, 0.16], and the other nine networks are set at [0.005, 0.04].



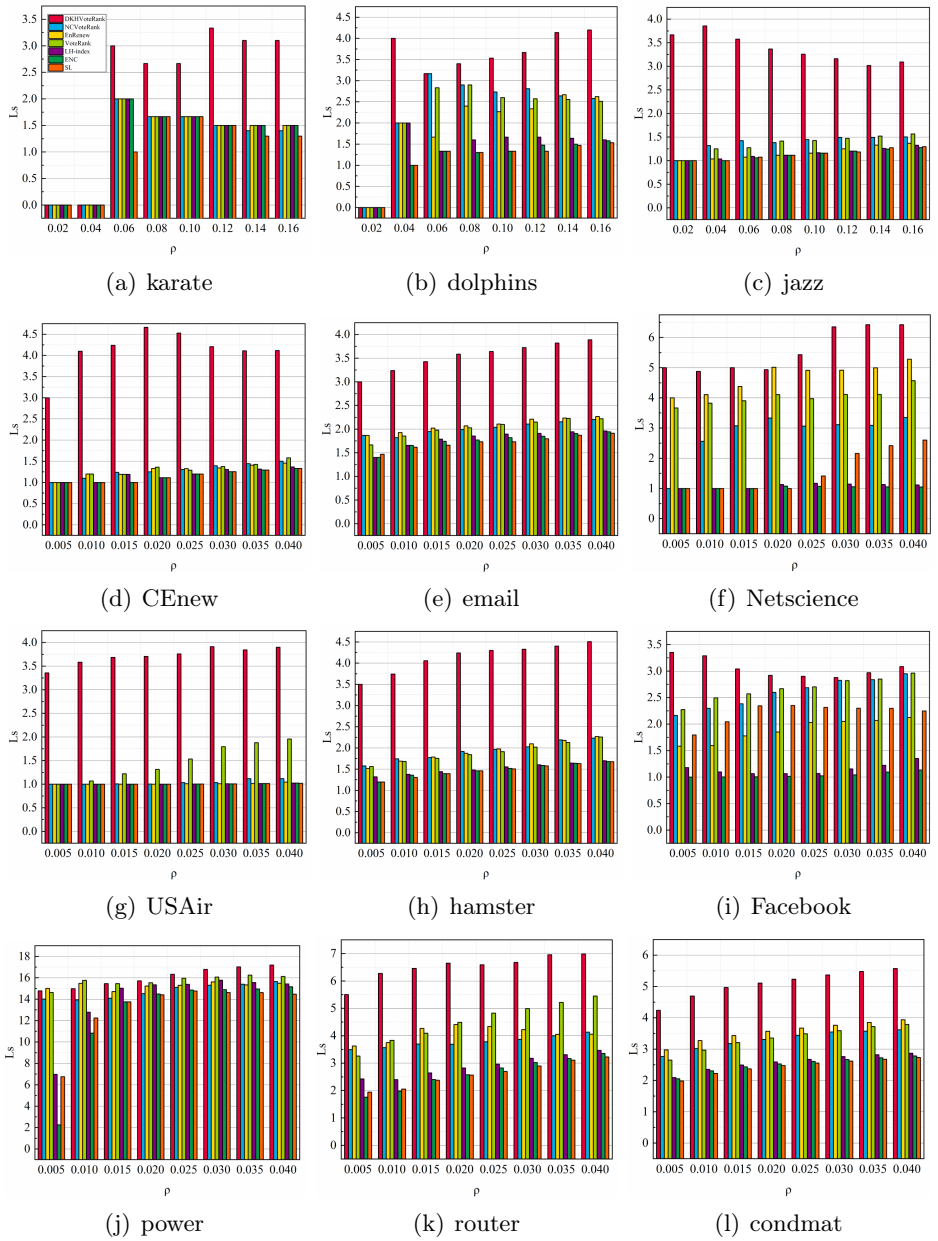
(a) karate (b) dolphins (c) jazz

(d) CEnew (e) email (f) Netscience

(g) USAir (h) hamster (i) Facebook

(j) power (k) router (l) condmat

Fig. 6. The variation curve of the final infection scale with the infection rate. We set the initial spreaders ratio $\rho = 0.02$, and take the average of 1000 simulations for each result. The node infection rate is another key indicator affecting the network propagation ability, and since we set the infection probability at $\beta = 1.5\beta_{\text{th}}$, we adjust the infection rate by changing the value of the node recovery probability during the experiment.



(a) karate (b) dolphins (c) jazz

(d) CEnew (e) email (f) Netscience

(g) USAir (h) hamster (i) Facebook

(j) power (k) router (l) condmat

Fig. 7. The average shortest distance of the initial spreaders identified by different algorithms. The average shortest path of the initial spreaders has an important impact on the final propagation effect. In order to make the initial spreaders to have a stronger influence on the network, we want the nodes identified by the algorithms to be spread out in different locations of the network as much as possible.

4.2.4. The average distance of different proportions of initial spreaders

As can be seen from Fig. 7, the initial spreaders identified by the DKHVoteRank algorithm have a higher distance, which means that these nodes are more dispersed in the network. The traditional importance ranking methods, such as DC, BC, CC, k-core, *etc.*, evaluate the importance of individual nodes without taking measures to avoid over-concentration of nodes, so the nodes identified by adopting the above algorithms may have a large clustering coefficient, which has an impact on the final propagation effect. The algorithm based on the voting method, after identifying the important nodes in the network, will weaken the voting ability of the neighbors of that node, thus avoiding the initial spreaders from being too clustered and affecting the propagation ability in the network. Our proposed algorithm, based on the traditional VoteRank algorithm, weakened the voting ability of all nodes with a distance of 2 from the selected node. It enhances the dispersion of the spreaders identified by this algorithm in the network even more. Therefore, the experimental results are significantly better than other algorithms.

5. Conclusion

In this paper, we propose an algorithm called DKHVoteRank to identify critical spreaders in complex networks. We optimize the VoteRank algorithm by introducing the DC, KC, and h-index methods, so that our method can better distinguish the importance of different nodes. At the same time, we improve the weakening rules of the VoteRank algorithm so that the critical spreaders identified by the DKHVoteRank algorithm are more widely distributed in the network compared to the traditional VoteRank algorithm. In order to compare the advantages and disadvantages of different algorithms, we perform simulation propagation in 12 different types of datasets based on the SIR model. According to the experimental results, our proposed algorithm has better performance in terms of propagation capability, propagation size, and algorithm applicability compared to VoteRank and its improved algorithms as well as improved algorithms of traditional classical ranking algorithms, such as NCVoteRank, EnRenew, ENC, SL, LH-index. In this paper, we verify the feasibility and effectiveness of our proposed algorithm in identifying critical spreaders in the network, which is valuable for limiting the propagation of information in the network and improving the destructive resistance of the network system.

REFERENCES

[1] S.H. Strogatz, «Exploring complex networks», *Nature* **410**, 268 (2001).

[2] S.P. Borgatti, A. Mehra, D.J. Brassand, G. Labianca, «Network analysis in the social sciences», *Science* **323**, 892 (2009).

[3] H. Dorussen, H. Ward, «Trade networks and the Kantian peace», *J. Peace Res.* **47**, 29 (2010).

[4] P. Zhang *et al.*, «The robustness of interdependent transportation networks under targeted attack», *Europhys. Lett.* **103**, 68005 (2013).

[5] R. Kinney, P. Crucitti, R. Albert, V. Latora, «Modeling cascading failures in the North American power grid», *Eur. Phys. J. B* **46**, 101 (2005).

[6] R. Sathyapriya, M.S. Vijayabaskar, S. Vishveshwara, «Insights into protein–DNA interactions through structure network analysis», *PLoS Comput. Biol.* **4**, e1000170 (2008).

[7] D. Chapman, B.V. Purse, H.E. Roy, J.M. Bullock, «Global trade networks determine the distribution of invasive non-native species», *Global Ecol. Biogeogr.* **26**, 907 (2017).

[8] J. Borge-Holthoefer, Y. Moreno, «Absence of influential spreaders in rumor dynamics», *Phys. Rev. E* **85**, 026116 (2012).

[9] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, A. Vespignani, «Epidemic processes in complex networks», *Rev. Mod. Phys.* **87**, 925 (2015).

[10] P. Liu, L. Li, S. Fang, Y. Yao, «Identifying influential nodes in social networks: A voting approach», *Chaos Solitons Fractals* **152**, 111309 (2021).

[11] B.A. Carreras, V.E. Lynch, «Critical points and transitions in an electric power transmission model for cascading failure blackouts», *Chaos* **12**, 985 (2002).

[12] R. Pastor-Satorras, A. Vespignani, «Epidemic spreading in scale-free networks», *Phys. Rev. Lett.* **86**, 3200 (2001).

[13] S.V. Buldyrev *et al.*, «Catastrophic cascade of failures in interdependent networks», *Nature* **464**, 1025 (2010).

[14] L.C. Freeman, «Centrality in social networks conceptual clarification», *Soc. Networks* **1**, 215 (1978).

[15] L.C. Freeman, «A set of measures of centrality based on betweenness», *Sociometry* **40**, 35 (1977).

[16] G. Sabidussi, «The centrality index of a graph», *Psychometrika* **31**, 581 (1966).

[17] D. Chen *et al.*, «Identifying influential nodes in complex networks», *Physica A* **391**, 1777 (2012).

[18] J. Liu *et al.*, «Evaluating the importance of nodes in complex networks», *Physica A* **452**, 209 (2016).

[19] Z.M. Ren *et al.*, «Node importance measurement based on the degree and clustering coefficient information», *Acta Phys. Sin.* **62**, 128901 (2013).

[20] M. Kitsak *et al.*, «Identification of influential spreaders in complex networks», *Nat. Phys.* **6**, 888 (2010).

[21] J.-G. Liu, Z.-M. Ren, Q. Guo, «Ranking the spreading influence in complex networks», *Physica A* **392**, 4154 (2013).

[22] M. Wang *et al.*, «Identifying influential spreaders in complex networks based on improved k-shell method», *Physica A* **554**, 124229 (2020).

[23] S. Yeruva *et al.*, T. Devi, Y. Samtha Reddy, «Selection of influential spreaders in complex networks using Pareto Shell decomposition», *Physica A* **452**, 133 (2016).

[24] J. Bae, S. Kim, «Identifying and ranking influential spreaders in complex networks by neighborhood coreness», *Physica A* **395**, 549 (2014).

[25] J.E. Hirsch, «An index to quantify an individual's scientific research output», *Proc. Natl. Acad. Sci. U.S.A.* **102**, 16569 (2005).

[26] Q. Liu *et al.*, «Leveraging local h-index to identify and rank influential spreaders in networks», *Physica A* **512**, 379 (2018).

[27] K. Bryan, T. Leise, «The 25,000,000,000 Eigenvector: The Linear Algebra behind Google», *SIAM Rev.* **48**, 569 (2006).

[28] G. Ghoshal, A.-L. Barabási, «Ranking stability and super-stable nodes in complex networks», *Nat. Commun.* **2**, 394 (2011).

[29] Q. Li, T. Zhou, L. Lv, D. Chen, «Identifying influential spreaders by weighted LeaderRank», *Physica A* **404**, 47 (2014).

[30] L. Page *et al.*, «The PageRank citation ranking: Bringing order to the web», Stanford InfoLab, 1999.

[31] T. Qiao, W. Shan, C. Zhou, «How to identify the most powerful node in complex networks? A novel entropy centrality approach», *Entropy* **19**, 614 (2017).

[32] A. Sheikhahmadi, M.A. Nematbakhsh, «Identification of multi-spreader users in social networks for viral marketing», *J. Inform. Sci.* **43**, 412 (2017).

[33] J.-G. Liu, Z.-M. Ren, Q. Guo, B.-H. Wang, «Node importance ranking of complex networks», *Acta Phys. Sin.* **62**, 178901 (2013).

[34] A. Zeng, C.-J. Zhang, «Ranking spreaders by decomposing complex networks», *Phys. Lett. A* **377**, 1031 (2013).

[35] J.-X. Zhang, D.-B. Chen, Q. Dong, Z.-D. Zhao, «Identifying a set of influential spreaders in complex networks», *Sci. Rep.* **6**, 27823 (2016).

[36] H.-L. Sun, D.-B. Chen, J.-L. He, E. Ch'ng, «A voting approach to uncover multiple influential spreaders on weighted networks», *Physica A* **519**, 303 (2019).

[37] S. Kumar, B. Panda, «Identifying influential nodes in social networks: neighborhood coreness based voting approach», *Physica A* **553**, 124215 (2020).

[38] C. Guo *et al.*, «Influential nodes identification in complex networks via information entropy», *Entropy* **22**, 242 (2020).

[39] A. Barrat, M. Barthelemy, A. Vespignani, «Dynamical processes on complex networks», Cambridge University Press, 2008.

[40] R. Pastor-Satorras, A. Vespignani, «Epidemic dynamics and endemic states in complex networks», *Phys. Rev. E* **63**, 066117 (2001).

[41] E.F. Codd, «A relational model of data for large shared data banks», in: M. Broy, E. Denert (Eds.) «Software pioneers. Contributions to Software Engineering», *Springer*, Berlin, Heidelberg 2002, pp. 263–294.

[42] Z.G. Liu *et al.*, «Node importance ranking of complex networks», *Acta Phys. Sin.* **62**, 178901 (2013).

[43] H.W. Hethcote, «The Mathematics of Infectious Diseases», *SIAM Rev.* **42**, 599 (2000).

[44] Z. Tao, F. Zhongqian, W. Binghong, «Epidemic dynamics on complex networks», *Prog. Nat. Sci.* **16**, 452 (2006).

[45] W.W. Zachary, «An information flow model for conflict and fission in small groups», *J. Anthropol. Res.* **33**, 452 (1977).

[46] D. Lusseau *et al.*, «The Bottlenose Dolphin Community of Doubtful Sound Features a Large Proportion of Long-lasting Associations», *Behav. Ecol. Sociobiol.* **54**, 396 (2003).

[47] P.M. Gleiser, L. Danon, «Community Structure in Jazz», *Adv. Complex Syst.* **06**, 565 (2003).

[48] H. Jeong *et al.*, «The large-scale organization of metabolic networks», *Nature* **407**, 651 (2000).

[49] R. Guimera *et al.*, «Self-similar community structure in a network of human interactions», *Phys. Rev. E* **68**, 065103 (2003).

[50] M.E. Newman, «Finding community structure in networks using the eigenvectors of matrices», *Phys. Rev. E* **74**, 036104 (2006).

[51] V. Colizza, R. Pastor-Satorras, A. Vespignani, «Reaction–diffusion processes and metapopulation models in heterogeneous networks», *Nat. Phys.* **3**, 276 (2007).

[52] K. Kunegis, «KONECT: the Koblenz Network Collection», in: «WWW'13 Companion: Proceedings of the 22$^{\text{nd}}$ International Conference on World Wide Web», Rio de Janeiro, Brazil May 13–17, 2013, pp. 1343–1350.

[53] J.J. McAuley, J. Leskovec, «Learning to Discover Social Sircles in Ego Networks», in: P. Bartlett (Ed.) «Advances in Neural Information Processing Systems 25 (NIPS 2012); 26$^{\text{th}}$ Annual Conference on Neural Information Processing Systems 2012», *Neural Information Processing Systems Foundation, Inc. (NeurIPS)*, USA 2013, pp. 539–548.

[54] D.J. Watts, S.H. Strogatz, «Collective dynamics of 'small-world' networks», *Nature* **393**, 440 (1998).

[55] N. Spring, R. Mahajan, D. Wetherall, «Measuring ISP topologies with rocketfuel», *ACM SIGCOMM Comput. Commun. Rev.* **32**, 133 (2002).

[56] J. Leskovec, J. Kleinberg, C. Faloutsos, «Graph evolution: Densification and shrinking diameters», *ACM Trans. Knowl. Discov. Data* **1**, 1 (2007).