

Article

The Structure Entropy-Based Node Importance Ranking Method for Graph Data

Shihu Liu  and Haiyan Gao *

School of Mathematics and Computer Science, Yunnan Minzu University, Kunming 650504, China; liush02@126.com

* Correspondence: 21213037550007@ymu.edu.cn

Abstract: Due to its wide application across many disciplines, how to make an efficient ranking for nodes in graph data has become an urgent topic. It is well-known that most classical methods only consider the local structure information of nodes, but ignore the global structure information of graph data. In order to further explore the influence of structure information on node importance, this paper designs a structure entropy-based node importance ranking method. Firstly, the target node and its associated edges are removed from the initial graph data. Next, the structure entropy of graph data can be constructed by considering the local and global structure information at the same time, in which case all nodes can be ranked. The effectiveness of the proposed method was tested by comparing it with five benchmark methods. The experimental results show that the structure entropy-based node importance ranking method performs well on eight real-world datasets.

Keywords: graph data; node importance ranking; structure entropy



Citation: Liu, S.; Gao, H. The Structure Entropy-Based Node Importance Ranking Method for Graph Data. *Entropy* **2023**, *25*, 941. <https://doi.org/10.3390/e25060941>

Academic Editor: Sotiris Kotsiantis

Received: 24 April 2023

Revised: 11 June 2023

Accepted: 13 June 2023

Published: 15 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As everyone knows, the key nodes usually play a decisive role during the process of graph data mining. In order to accurately identify the so-named key nodes in graph data, a priority problem is to construct an appropriate score function for ranking nodes [1–4]. Due to its prevalence in the field of disease detection [5,6], information transmission [7,8] and rumor blocking [9,10], how to rank nodes in graph data has been widely studied by researchers of various vocations.

In general, there are many traditional node importance ranking methods that only considered the local structure information of nodes to construct the score function [11–13]. For example, Lu et al. [14] calculated the importance of nodes by means of the degree centrality method. Chen et al. [15] constructed a multi-level neighbor information index to measure the importance of nodes, in which case only the degree information of first-order and second-order neighbors are considered. In order to distinguish the contribution of different neighbors, Katz [16] assigned different weights to them. The neighbors that can be reached by the short route are assigned the larger weight. At the same time, the neighbors that can be reached by the long route are assigned the small weight [17].

Up to now, many improved methods are proposed to deal with the problem of node importance ranking for graph data [18–20]. For instance, Freeman [21] constructed the betweenness centrality method, which described the importance of a node as the number of the shortest paths through it. In the closeness centrality method [22], the importance score of each node can be determined through the impact ability of the target node on other nodes. Based on the hypothesis that the node located at the core position has a strong influence, the K-shell decomposition method [23] is proposed. Yang et al. [24] proposed a comprehensive evaluation method based on multi-attribute decision making, which took many factors that affect the importance of nodes into account. What is more, the graph learning framework is also applied to evaluate the importance of nodes, such as

in reference [25], the first graph neural network-based model is proposed to approximate betweenness and closeness centrality. Furthermore, Liu et al. [26] proposed a novel model based on self-supervised learning and graph convolution model to rank nodes, which formulated node importance ranking problem as a learning ranking problem.

Besides what has been discussed above, the well-known information entropy that has been proposed by Shannon [27] is also regarded as a powerful tool to measure the importance of nodes in a whole new perspective [28–30]. For instance, Zareie et al. [31] constructed the score function for each node, which considered the influence of neighbors on the target node with the help of information entropy. Guo et al. [32] proposed the EnRenew method by using the voting mechanism. In this method, information entropy is regarded as the voting ability of neighbors. By taking the effect of the spreading rate on information entropy into account, a propagation feature of the node-based ranking approach is introduced in reference [33]. Yu et al. [34] characterized the node importance as the node propagation entropy, which was the combination of degree and clustering coefficients.

Based on the above analysis, it can be found easily that in both the information entropy-based ranking methods and traditional ranking methods, only the local structure information is used to construct score functions. However, in fact, the global structure information, i.e., the connectivity of whole graph data, usually has a huge influence on the final ranking sequence [35–37]. In order to overcome the limitation or make full use of information from graph data, we propose a structural entropy-based node importance ranking method by considering the global structure information of graph data. We first calculate the amount of information contained in each connected component, which is denoted as the local structural entropy. Furthermore, the global structure entropy is constructed by distinguishing the different contributions of each connected component. Moreover, the effectiveness of the proposed method was tested on eight real-world datasets. The contribution of this paper can be listed as follows.

- The structure entropy-based node importance ranking method for graph data is proposed in terms of node removal.
- The local structural entropy is calculated by considering the degree of information of nodes and information entropy.
- The global structure entropy is constructed in terms of the connectivity of graph data.

The remainder of this paper is organized as follows. Section 2 reviews some basic concepts, which are graph data and benchmark methods for node importance ranking. Section 3 introduces the proposed method, i.e., the structural entropy-based node importance ranking method. Section 4 is composed of three parts, which are the experimental platform, datasets description and evaluation criteria. Section 5 shows the experimental results and contrastive analysis between the proposed method and five benchmark methods on eight real-world datasets. Section 6 is the summary of this paper and gives future research directions.

2. Preliminaries

In this section, some basic concepts are introduced, including the graph data and some benchmark methods for node importance ranking [38–41].

2.1. Graph Data

Generally speaking, the so-called graph data can be expressed as a tuple $G = (V, E)$, where

- $V = \{v_i | i = 1, 2, \dots, n\}$ is the set of nodes and n represents the number of nodes.
- $E = \{(v_i, v_j) | v_i, v_j \in V\}$ is the set of edges and $m = |E|$ represents the number of edges.

In this paper, we mainly discuss the undirected and unweighted graph data G . That is to say, $(v_i, v_j) = (v_j, v_i)$ for any $v_i, v_j \in V$. Given that $v_i, v_j \in V$, $(v_i, v_j) \in E$ if and only if

there exists one edge that takes nodes v_i and v_j as its endpoint. For this situation, we use $a_{ij} = 1$ to describe the fact that v_i and v_j are adjacent. Similarly, $a_{ij} = 0$ denotes that v_i and v_j are not adjacent. With this representation, the adjacency of a given graph data G with n nodes is the following matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}. \quad (1)$$

2.2. Benchmark Methods

The key problem of node importance ranking is how to construct the score function. It is well-known that most classical methods apply local structure information of nodes to construct score functions. Some benchmark methods that can be used to rank the nodes are introduced in what follows.

2.2.1. Degree Centrality Method

The degree centrality method (**DC**) determines the importance of node v_i by the following equation

$$DC(v_i) = d_i, \quad (2)$$

where $d_i = \sum_{j=1}^n a_{ij}$ is the degree of node v_i .

2.2.2. Closeness Centrality Method

The closeness centrality method (**CC**) defines the importance of node v_i is

$$CC(v_i) = \frac{1}{\sum_{i \neq j} d(i, j)}, \quad (3)$$

where $d(i, j)$ is the length of the shortest path from node v_i to v_j , or v_j to v_i .

2.2.3. Improved K-Shell Decomposition Method

The classical K-shell decomposition method (**KS**) is a node removal-based method. A different K_s value that is regarded as the corresponding importance score is assigned to different nodes. In the first place, nodes with $d_i \leq 1$ are removed from the initial graph data G , and the same time value of $K_s = 1$ is assigned to such nodes. After that, for the newly generated graph data, nodes with $d_i \leq 2, 3, \dots$, will be removed successively, in which case one will obtain the sequence $K_s = 2, 3, \dots$. For the improved K-shell decomposition method (**IKS**), it only removes nodes with the lowest degree each iteration. That is to say, the sequence of removed nodes is not based on the increasing sequence of degrees. For example, when all nodes with $d_i = 2$ are removed, the node with $d_i = 1$ may appear in the newly generated graph data. These nodes with $d_i = 1$ will be removed next and obtain a higher IK_s value.

2.2.4. The Weight of Edges-Based Method

The weight of edges-based method (**WR**) determines the importance of node v_i by the following equation

$$WR(v_i) = \sum_{v_j \in N(v_i)} d_i d_j, \quad (4)$$

where $N(v_i) = \{v_j | (v_i, v_j) \in E\}$ is the set of neighbors of node v_i .

2.2.5. The Gravity Model Based Method

Inspired by the thought of the classical gravity model, the gravity model-based method (*GM*) quantifies the importance of nodes by combining *Ks* value and shortest path information of nodes. The concrete calculation formula is

$$GM(v_i) = \sum_{v_j \in \Psi(v_i)} \frac{Ks(v_i)Ks(v_j)}{d(i, j)^2}, \tag{5}$$

where $\Psi(v_i)$ is the set of nodes that defined by equation $\Psi(v_i) = \{v_j \mid d(i, j) \leq 3\}$.

3. The Proposed Method

It is well-known that most of the classical node importance ranking methods only consider the local structure information of nodes, but ignore the global structure information of graph data. For this, we combine the local and global structure information to construct the score function for all nodes. Based on the assumption that removing a more important node is likely to cause more structural variation of graph data, the score function is constructed from the perspective of node removal. Furthermore, the local and global structure information are considered comprehensively to construct the structure entropy of graph data and in which case all nodes can be ranked.

3.1. Node Removal

The graph data $G = (V, E)$ are defined as a connected graph if there is a route from v_i to v_j , or v_j to v_i for any nodes $v_i, v_j \in G$. Otherwise, it is a disconnected graph. For a disconnected graph, each connected part is called a connected component.

Taking Figure 1, for example, there are 12 nodes and 14 edges. One can find that the nodes v_3 and v_5 have the same degree. They will be assigned the same importance score according to the *DC* method. However, in fact, the importance of these two nodes is completely different.

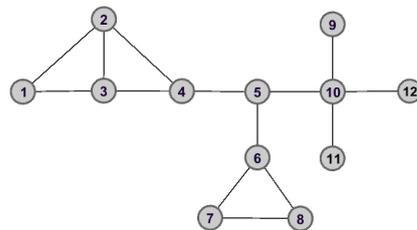


Figure 1. The connected graph with 12 nodes and 14 edges.

As shown in Figures 2 and 3, the graph data are divided into three connected components when node v_5 is removed. However, the removal of node v_3 does not lead to great changes for the structure of graph data, and the remaining graph data is still connected. Therefore, we can make the assertion that node v_5 would play a more important role than that of node v_3 in the aspect of structure information.

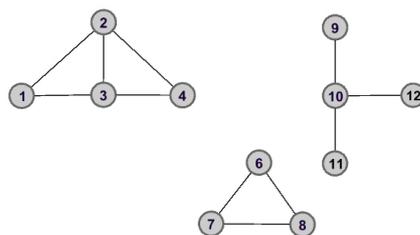


Figure 2. The graph data after removing node v_5 .

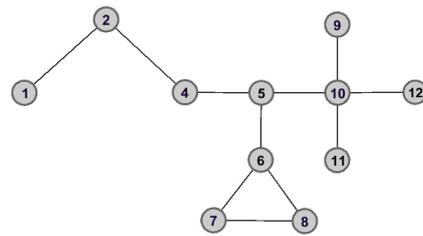


Figure 3. The graph data after removing node v_3 .

3.2. Local Structure Entropy

Removing the important node may lead to the fact that the graph data will be divided into more than one connected component. In order to quantify the global structure information of graph data reasonably, calculating the amount of information about each connected component is a priority problem. Hereinto, we first construct the local structure entropy for each connected component with the help of information entropy.

The information entropy is usually used to measure the amount of information about an event. For the random variable $X = (x_1, x_2, \dots, x_n)$, given that its probability distribution is $P = (p_1, p_2, \dots, p_n)$, then the information entropy of X is

$$E(X) = - \sum_{i=1}^n p_i \log_2 p_i. \tag{6}$$

Following Equation (6), one can find that the more likely an event is to happen, the less information it contains, and vice versa. Once some nodes are removed, the graph data will be changed into more than one connected component with a high probability. This will lead to information decreasing of the corresponding connected component. That is to say, the appearance of connected components is a frequent event and it contains less structure information. This is consistent with the property of information entropy. Therefore, the amount of structure information contained in each connected component can be quantified by information entropy, and it can be defined as local structural entropy. In what follows, we give a detailed description of local structural entropy.

Given that $G = (V, E)$ is graph data with n nodes and m edges. The initial graph data are divided into s connected components after removing the target node from G , denoted as C_1, C_2, \dots, C_s . Each connected component contains $|C_i|$ nodes, for $i = 1, 2, \dots, s$. Then, the probability distribution $P(C_i)$, for $i = 1, 2, \dots, s$, can be expressed as

$$P(C_i) = (p(v_1), p(v_2), \dots, p(v_{|C_i|})), \tag{7}$$

where

$$p(v_t) = \begin{cases} \frac{\sum_{v_j \in N(v_t)} d_j}{\sum_{v_x \in C_i} d_x^2}, & |C_i| > 1 \\ 1, & |C_i| = 1 \end{cases} \tag{8}$$

for $t = 1, 2, \dots, |C_i|$. Obviously, this probability distribution satisfies the constraint that the sum of probability is equal to 1 for each connected component, i.e., $\sum_{v_t \in C_i} p(v_t) = 1$.

According to Equation (6), the local structure entropy with respect to the connected component C_i , for $i = 1, 2, \dots, s$, can be defined as

$$LE(C_i) = - \sum_{v_j \in C_i} p(v_j) \log_2 p(v_j). \tag{9}$$

It can be easily found that Equation (9) has the following properties.

Property 1. Given that G is graph data and C_i is the i th connected component of G by removing v_i from G . Then, one has that $LE(C_i) \geq 0$.

Proof. If $|C_i| = 1$, taking $v_t \in C_i$ for example, then $p(v_t) = 1$. To this

$$\begin{aligned} LE(C_i) &= - \sum_{v_j \in C_i} p(v_j) \log_2 p(v_j) \\ &= p(v_t) \log_2 p(v_t) \\ &= 0. \end{aligned} \tag{10}$$

If $|C_i| > 1$, one has that $p(v_j) \log_2 p(v_j) < 0$ for any $v_j \in C_i$, then

$$\begin{aligned} LE(C_i) &= - \sum_{v_j \in C_i} p(v_j) \log_2 p(v_j) \\ &> 0. \end{aligned} \tag{11}$$

This completes the proof. \square

Property 2. Given that G is a graph data and C_i is the i th connected component of G by removing v_i from G . Then, the value of $LE(C_i)$ is not relevant to the position of $p(v_t)$ in $P(C_i)$, for $v_t \in C_i$.

Proof. For the connected component C_i , the initial probability distribution is $P(C_i) = (p(v_1), p(v_2), \dots, p(v_{|C_i|}))$. If $p(v_1)$ and $p(v_2)$ change the position, the probability distribution changes into $P(\bar{C}_i) = (p(v_2), p(v_1), \dots, p(v_{|C_i|}))$.

With the help of Equation (9), the following result

$$\begin{aligned} LE(C_i) &= - \sum_{v_j \in C_i} p(v_j) \log_2 p(v_j) \\ &= -(p(v_1) \log_2 p(v_1) + p(v_2) \log_2 p(v_2)) - \sum_{v_j \in C_i, v_j \neq v_1, v_2} p(v_j) \log_2 p(v_j) \\ &= -(p(v_2) \log_2 p(v_2) + p(v_1) \log_2 p(v_1)) - \sum_{v_j \in C_i, v_j \neq v_1, v_2} p(v_j) \log_2 p(v_j) \\ &= LE(\bar{C}_i) \end{aligned} \tag{12}$$

comes naturally.

This completes the proof. \square

Property 3. Given that C_i and C_j , respectively, are the i th and j th connected components of G by removing node v_i from G . Then, their overall structure entropy can be expressed as the sum of local structure entropy, i.e., $LE(C_i C_j) = LE(C_i) + LE(C_j)$.

Proof. According to the Equations (7) and (8), the probability distributions of C_i and C_j is

$$P(C_i) = (p(v_1), p(v_2), \dots, p(v_{|C_i|})) \tag{13}$$

and

$$P(C_j) = (p(v'_1), p(v'_2), \dots, p(v'_{|C_j|})), \tag{14}$$

where $\sum_{v_t \in C_i} p(v_t) = 1$, for $t = 1, 2, \dots, |C_i|$ and $\sum_{v'_x \in C_j} p(v'_x) = 1$, for $x = 1, 2, \dots, |C_j|$.

For independent connected components C_i and C_j , their joint probability distribution can be expressed as

$$P(C_i C_j) = (p(v_1)p(v'_1), p(v_1)p(v'_2), \dots, p(v_1)p(v'_{|C_j|}), p(v_2)p(v'_1), \dots, p(v_{|C_i|})p(v'_{|C_j|})),$$

where $\sum_{v_t \in C_i} \sum_{v'_x \in C_j} p(v_t)p(v'_x) = 1$, for $t = 1, 2, \dots, |C_i|$ and $x = 1, 2, \dots, |C_j|$.

With the help of Equation (9), one can have that

$$\begin{aligned} LE(C_i C_j) &= - \sum_{v_t \in C_i} \sum_{v'_x \in C_j} p(v_t)p(v'_x) \log_2(p(v_t)p(v'_x)) \\ &= - \sum_{v_t \in C_i} \sum_{v'_x \in C_j} p(v_t)p(v'_x) \log_2 p(v_t) - \sum_{v_t \in C_i} \sum_{v'_x \in C_j} p(v_t)p(v'_x) \log_2 p(v'_x) \\ &= - \sum_{v'_x \in C_j} p(v'_x) \sum_{v_t \in C_i} p(v_t) \log_2 p(v_t) - \sum_{v_t \in C_i} p(v_t) \sum_{v'_x \in C_j} p(v'_x) \log_2 p(v'_x) \\ &= LE(C_i) + LE(C_j). \end{aligned} \tag{15}$$

This completes the proof. \square

3.3. Global Structure Entropy

The key problem in this section is to quantify the information contained in the whole graph data. According to Property 3, one can find that the overall structure entropy of G can be expressed as the sum of local structure entropy about each connected component. In order to distinguish the contribution of different connected components, in what follows, we take the number of edges as the weight value of each local structure entropy.

Given that G is graph data and taking node $v_i \in V$ as an example, the global structure entropy of G is quantified by combining the number of edges and local structure entropy, which can be defined as

$$SE(v_i) = \sum_{j=1}^s |E_j| LE(C_j), \tag{16}$$

where $|E_j|$ is the number of edges in each connected component, for $j = 1, 2, \dots, s$.

The information contained in graph data G will decrease if the more important node v_i is removed. That is to say, the global structure entropy will get the smaller value of $SE(v_i)$. Therefore, the global structure entropy can be regarded as a cost function. In other words, the smaller the value of $SE(v_i)$, the more important the node v_i . For this, one can obtain a possible sequence, such as $v_{i1} \succcurlyeq v_{i2} \succcurlyeq \dots \succcurlyeq v_{in}$, where $(i1, i2, \dots, in)$ is a certain permutation of $(1, 2, \dots, n)$. For example, $v_{i1} \succcurlyeq v_{i2}$ if and only if $SE(v_{i1}) \leq SE(v_{i2})$, and $v_{i1} \prec v_{i2}$ if and only if $SE(v_{i1}) > SE(v_{i2})$.

Example 1. To make it easy to understand how to calculate the global structure entropy of each node, in what follows, we apply a simple graph data G shown in Figure 1 to describe the whole process in detail.

Taking node v_5 for example, the initial graph data is divided into three connected components after removing node v_5 from G , which are C_1, C_2 and C_3 . With the help of Equations (7) and (8), the probability distribution of connected components can be determined, which are

$$P(C_1) = \left(\frac{6}{26}, \frac{7}{26}, \frac{7}{26}, \frac{6}{26} \right)$$

$$P(C_2) = \left(\frac{4}{12}, \frac{4}{12}, \frac{4}{12} \right)$$

and

$$P(C_3) = \left(\frac{3}{12}, \frac{3}{12}, \frac{3}{12} \right).$$

Then, the local structure entropy of each connected component could be obtained by Equation (9), which are

$$LE(C_1) = 1.9958$$

$$LE(C_2) = 1.5849$$

and

$$LE(C_3) = 2.0000.$$

With Equation (16), the global structure entropy is

$$SE(v_5) = \sum_{j=1}^3 |E_j|LE(C_j) = 20.7337. \tag{17}$$

The calculation of other nodes is the same as that of v_5 . Here, we list the top six nodes in Table 1.

Table 1. The global structure entropy of top six nodes.

Order	1	2	3	4	5	6
Node	v_5	v_4	v_{10}	v_6	v_2	v_3
$SE(v_i)$	20.7337	28.5322	29.7450	32.1098	36.9700	36.9700

As can be seen from Table 1, one has that $SE(v_5) < SE(v_3)$, then their importance can be ranked as $v_5 \succ v_3$. It is worth mentioning that this is consistent with the analysis results in Section 3.1.

3.4. Algorithm Description

Bearing what was discussed in mind, we give the detailed process of the structure entropy-based node importance ranking method for graph data G in Algorithm 1. For convenience, here we apply the abbreviation SE to represent the proposed method.

Algorithm 1: The SE method.

```

input : The undirected and unweighted graph data  $G = (V, E)$ .
output: The ranking sequence  $v_{i1} \succ v_{i2} \succ \dots \succ v_{in}$ .

1 begin
2   for  $i = 1 : n$  do
3      $\bar{V} \leftarrow V \setminus \{v_i\}$ ;
4      $\bar{E} \leftarrow E \setminus \{(v_i, v_j) | v_j \in N(v_i)\}$ ;
5     for  $k = 1 : s$  do
6       | Compute local structure entropy  $LE(C_k)$  by Equation (9);
7     end
8     | Compute global structure entropy  $SE(v_i)$  by Equation (16);
9   end
10  for  $v_i, v_j \in V$  do
11    | if  $SE(v_i) \leq SE(v_j)$ 
12      |    $v_i \succ v_j$ ;
13    | else
14      |    $v_i \prec v_j$ ;
15    | end
16  end
17  return  $v_{i1} \succ v_{i2} \succ \dots \succ v_{in}$ .           /*  $v_{ij} \in V$ , for  $j = 1, 2, \dots, n$  */
18 end

```

4. Experimental Construction

In this section, we introduce the experimental platform, experimental datasets and evaluation criteria.

4.1. Experimental Platform

The algorithm development platform is MATLAB R2018a. The computer configuration used for the experiment is the following: Intel(R)Core(TM)i5-8250U CPU, 8 GB installed memory and 64-bit Windows 10 operating system.

4.2. Datasets Description

From the website <http://konect.cc/networks/> (accessed on 10 April 2023), we downloaded the eight real-world datasets for experimental analysis. The detailed information on these datasets is given below.

- **Contiguous USA (CONT):** The network of shared border between 48 contiguous states.
- **Les Miserables (LESM):** The network of co-appearances of characters in the novel “Les Miserables”.
- **Polbooks (POLB):** The network of books about US politics published in 2004.
- **Adjnoun (ADJN):** The network of co-words between adjectives and nouns commonly used in the novel “David Copperfield”.
- **Football (FOOT):** The network of US football games between division IA colleges.
- **Netscience (NETS):** The collaborative network of scientists who have published papers in the field of network science.
- **Email (EMAI):** The interactive network of emails between members in University of Rovira.
- **Hamsterster households (HAMS):** The network of family relationships between members using the same website.

Table 2 shows the topological statistical information of the above eight real-world datasets, where $\langle d \rangle$ is the average degree, d_{max} is the maximum degree and cc is the average clustering coefficient of datasets ($cc = \frac{1}{n} \sum_{i=1}^n c_i$, where c_i is the local clustering

coefficient of node v_i and $c_i = \frac{\sum_{v_j \in N(v_i)} d_j}{d_i(d_i-1)}$).

Table 2. The topological statistical information of eight real-world networks.

Dataset	n	m	$\langle d \rangle$	d_{max}	cc
CONT	49	107	4.3673	8	0.4061
LESM	77	254	6.5974	36	0.4989
POLB	105	441	8.4000	25	0.4875
ADJN	112	425	7.5893	49	0.1898
FOOT	115	613	10.6609	12	0.4032
NETS	379	914	4.8232	34	0.7981
EMAI	1133	10,903	9.6230	71	0.2550
HAMS	1576	4032	5.1168	147	0.1312

As shown in Table 2, the eight real-world datasets used for the experimental analysis have the following different properties. The number of nodes in CONT and LESM datasets are both less than 100, which is mainly used to verify the effectiveness of the proposed method on small-scale datasets. Although POLB, ADJN and FOOT datasets have similar scale, $\langle d \rangle$ and d_{max} of the FOOT dataset are very close. This indicates the fact that there are a large number of nodes with the same degree in the FOOT dataset. Since the NETS dataset has the largest cc in all datasets, the distribution of nodes is dense. The EMAI and HAMS belong to the larger-scale datasets. Hereinto, the EMAI is the dataset with the

highest number of edges. The *HAMS* dataset has the highest number of nodes and the smallest average clustering coefficient in all datasets. In fact, the biggest difference between the *HAMS* and other datasets is that it contains 655 isolated nodes. The extensibility of ranking methods can be reflected in this kind of special dataset.

4.3. Evaluation Criteria

Here, we introduce four evaluation criteria to verify the validity of the proposed method. The more detailed information can be found in the literature [42–45].

4.3.1. Monotonicity-Based Evaluation Criterion

It is well-known that the fewer nodes that obtain the same importance score, the better the corresponding ranking method. Here, the discriminability of the proposed method can be evaluated by using the monotonicity relation function. Its mathematical formula is

$$M(R) = \left(1 - \frac{\sum_{r \in \Gamma} n_r(n_r - 1)^2}{n(n - 1)} \right)^2, \tag{18}$$

where R is the final ranking sequence, Γ is the index set that represents different orders in the ranking sequence R , $r \in \Gamma$ and n_r represents the number of nodes that have been listed in the same order. For example, if the ranking sequence R is $v_1 \succ v_2 \approx v_3 \succ v_4$, then $\Gamma = \{1, 2, 3\}$ and $n_1 = n_3 = 1$ and $n_2 = 2$. Obviously, if all nodes have the same order in the ranking sequence R , then the value of $M(R)$ is 0. If each node can obtain a unique order, the value of $M(R)$ is 1 and the ranking sequence R is completely monotonic.

4.3.2. Complementary Cumulative Distribution Function Based Evaluation Criterion

In addition to monotonicity, the complementary cumulative distribution function (*CCDF*) is utilized to further evaluate the discriminability of the proposed method. Its mathematical expression is

$$CCDF(r) = \frac{n - \sum_{i=1}^r n_i}{n}. \tag{19}$$

Obviously, with the increasing of r , if more nodes are assigned to the same order, then the value of the function will decrease rapidly, until to 0.

4.3.3. Connected Component Based Evaluation Criterion

Generally, the robustness of the ranking method can be quantified by the deliberate attack strategy. Firstly, some nodes are removed from graph data G according to the ranking sequence R , which can decrease the connectivity of G . After that, the robustness of the ranking method is evaluated from two perspectives, i.e., the number of connected components and the proportion of the maximum connected component. The former can be expressed as ζ , and the definition of latter is

$$\tau = \frac{M_s}{n}, \tag{20}$$

where

$$M_s = \max\{|C_1|, |C_2|, \dots, |C_s|\} \tag{21}$$

represents the number of nodes that are contained in the maximum connected component. Obviously, one can find that the larger value of ζ and the smaller value of τ , the stronger the robustness of the corresponding ranking method.

4.3.4. Susceptible-Infected-Recovered Epidemic Model-Based Evaluation Criterion

The accuracy of different ranking methods can be verified by using the Susceptible-Infected-Recovered epidemic model (*SIR*). Nodes in *SIR* are classified into infected state, susceptible state, and recovered state. In the whole process of infection, the initial infected node can affect its neighbors with the infected probability β , and enter into a recovered state with the recovery probability γ . Nodes that are already in the recovery state will not participate in the subsequent infection process. To increase accuracy, the experiment will repeat hundreds of times and the average number of infected nodes is taken as the propagation ability of the seed node, denoted as $F(R)$. Its calculation formula is defined as

$$F(R) = \frac{n_I}{N_{ite}}, \tag{22}$$

where n_I is the number of nodes infected by seeds and N_{ite} is the number of repeated experiments.

5. Results and Analysis

In this section, the performance of the proposed method *SE* is demonstrated on eight real-world datasets. In order to show the results more clearly, all datasets are classified into three classes in the aspect of the number of nodes, i.e., the datasets *CONT* and *LESM* with $n \leq 100$, the datasets *POLB*, *ADJN*, *FOOT* and *NETS* with $100 < n \leq 1000$, the datasets *EMAI* and *HAMS* with $n > 1000$.

5.1. Monotonicity Analysis

In this part, we analyze the effectiveness of *SE* by comparing the monotonicity of ranking sequence *R* obtained by *SE* with other benchmark methods. Table 3 shows the value of monotonicity under *DC*, *CC*, *IKS*, *WR*, *GM* and *SE* methods. One can find that the *SE* method can obtain the maximum monotonicity value on all datasets. Obviously, this advantage is independent of the number of nodes.

Table 3. The M value of six ranking methods. The best results are highlighted in bold.

Dataset	M (DC)	M (CC)	M (IKS)	M (WR)	M (GM)	M (SE)
<i>CONT</i>	0.6973	0.9780	0.7942	0.9546	0.9966	1.0000
<i>LESM</i>	0.8147	0.9414	0.8134	0.9547	0.9581	0.9581
<i>POLB</i>	0.8252	0.9846	0.8382	0.9967	0.9996	1.0000
<i>ADJN</i>	0.8661	0.9837	0.8745	0.9961	0.9994	0.9997
<i>FOOT</i>	0.3636	0.9488	0.9419	0.9281	0.9985	1.0000
<i>NETS</i>	0.7642	0.9928	0.7607	0.9839	0.9949	0.9953
<i>EMAI</i>	0.8874	0.9988	0.8981	0.9977	0.9999	0.9999
<i>HAMS</i>	0.6263	0.6834	0.6292	0.6829	0.6839	0.6839

5.1.1. On *CONT* and *LESM* Datasets

From Table 3, one can find that for the *CONT* dataset, all methods except *DC* and *IKS*, the monotonicity is greater than 0.9000. The main reason is that the two methods, i.e., *DC* and *IKS* methods, can be influenced easily by the degree of nodes. It is worth mentioning that the *SE* method is less affected by the degree of information about nodes. Therefore, it is superior to *DC* and *IKS* methods in monotonicity.

On the *LESM* dataset, it should be pointed out that both *SE* and *GM* methods can achieve the maximum value of monotonicity at the same time. From Table 2, one can find that the *LESM* dataset has a higher *cc* value in datasets with a similar number of nodes. For datasets with dense distribution of nodes, the method that the structure information of nodes is considered during the ranking procedure can identify the importance of nodes more efficiently, such as *SE* and *GM* methods. This also confirms that the *SE* method has great merit on small-scale datasets.

5.1.2. On *POLB*, *ADJN*, *FOOT* and *NETS* Datasets

Since the *POLB*, *ADJN* and *FOOT* datasets have similar scales, most methods achieve similar monotonicity. In this case, the *SE* method still shows obvious advantages. One can observe that the *SE* method not only obtains the highest monotonicity value on all datasets but also assigns the unique order to each node on *POLB* and *FOOT* datasets. Table 2 shows that $\langle d \rangle$ and d_{max} of the *FOOT* dataset are very close. This indicates the fact that there are a large number of nodes with the same degree in the *FOOT* dataset. Since the *DC* method have no ability to identify the importance of these nodes, it achieves the worst monotonicity. On the contrary, the *SE* method can obtain a completely monotonous ranking sequence. What is more, the difference in monotonicity value between *SE* and *DC* methods is as high as 0.6464. For this, we can guess that the *SE* method would show better performance on large-scale graph data.

On the *NETS* dataset, the *CC* and *GM* methods obtain similar monotonicity values, but *IKS* is still the worst-performing method. Since the *NETS* dataset has the largest *cc* in all datasets, the distribution of nodes is dense. Obviously, the *IKS* method has the worst performance on this dataset. The main reason is that the *IKS* method mainly considers the location information of nodes largely, and usually treats nodes with adjacent locations as equally important. On the contrary, the *SE* method is not affected by the location of nodes and can still obtain the maximum value of monotonicity.

5.1.3. On *EMAI* and *HAMS* Datasets

For datasets with a large number of nodes, such as the *EMAI* and *HAMS* datasets, one can find that the *GM* method shows the same advantage as the *SE* method and the performance of *CC* and *WR* methods also increases. As shown in Table 2, the *HAMS* dataset has the highest number of nodes and the smallest clustering coefficient in all datasets, which indicates that the nodes in the *HAMS* dataset are more dispersed. In fact, the biggest difference between the *HAMS* and other datasets is that it contains 655 isolated nodes. Due to this special structure, the importance of most nodes cannot be identified on the *HAMS* dataset. However, the *SE* method still obtains the maximum value of monotonicity. This further verifies the effectiveness of the *SE* method on datasets with special structure.

5.2. Node Distribution Analysis

As shown in Figures 4–6, the *CCDF* curves express the node distribution of the ranking sequence obtained by different methods. Here, we mainly focus on two perspectives. On the one hand, the descending slope of curves can indicate the discriminability of the corresponding method. The method with the smoother descending slope can distribute the fewer nodes in the same order. On the other hand, we focus on the value of the horizontal axis when the value of the vertical axis is equal to 0, which can represent the total order number that can be generated by the corresponding method. The larger the order number, the better the discriminability of the corresponding method.

From the results, it can be found that the *SE* method can obtain the smoother descending slope and the maximum order number on most datasets. That is to say, the *SE* method should distribute the fewer nodes to the same order and more clearly identify the importance of different nodes compared with benchmark methods.

5.2.1. On *CONT* and *LESM* Datasets

Figure 4 is the curves of *CCDF* on *CONT* and *LESM* datasets. Obviously, the *SE* method obtains the smoothest descending slope and descends keeping in a straight line as shown in Figure 4a. What is more, the total order number obtained by the *SE* method is 49, which is equivalent to the node number of *CONT* dataset. In other words, only one node is located at the corresponding location of the ranking sequence. Nicely, this is consistent with that of Table 3.

From Figure 4b, it can be easily found that both *GM* and *SE* methods obtain the maximum order number 52 at the same time. However, that of *DC* and *IKS* methods is

18, which means that there are 59 nodes whose importance cannot be identified. Such defects are more evident on larger-scale datasets and this can be confirmed by the following experiments. In addition, although there is no method that can completely identify the importance of all nodes, the *SE* method shows greater advantage when the order number is between 10 and 20.

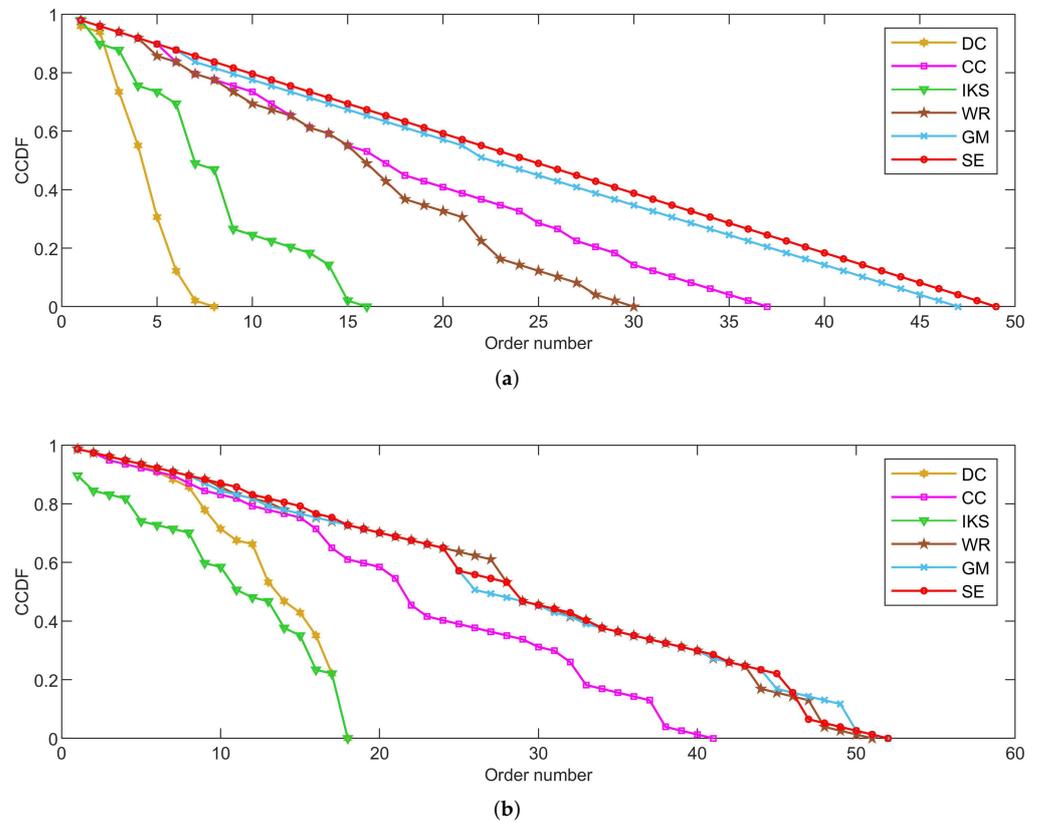


Figure 4. The curves of *CCDF* on (a) *CONT* and (b) *LESM* datasets.

5.2.2. On *POLB*, *ADJN*, *FOOT* and *NETS* Datasets

Figure 5 is the curves of *CCDF* on *POLB*, *ADJN*, *FOOT* and *NETS* datasets. As can be seen, the advantage of the *SE* method is obvious. On the one hand, the *SE* method achieves the maximum order number on all datasets. This reflects the fact that the *SE* method can distribute fewer nodes to the same order compared with other benchmark methods. On the other hand, the *SE* method obtains the smoothest descending slope in all of the comparison methods. Especially on the *POLB* and *FOOT* datasets, *SE* is the only method that can descend keeping in a straight line. For this, we can guess that the *SE* method would show better performance on large-scale datasets.

On the whole, the performance of *DC* and *IKS* methods is relatively poor. The order number obtained by *DC* and *IKS* methods is only 10% to 20% of the total number of nodes. This means that nearly 80% to 90% of nodes' importance cannot be identified. Frankly speaking, they cannot be regarded as good ranking methods.

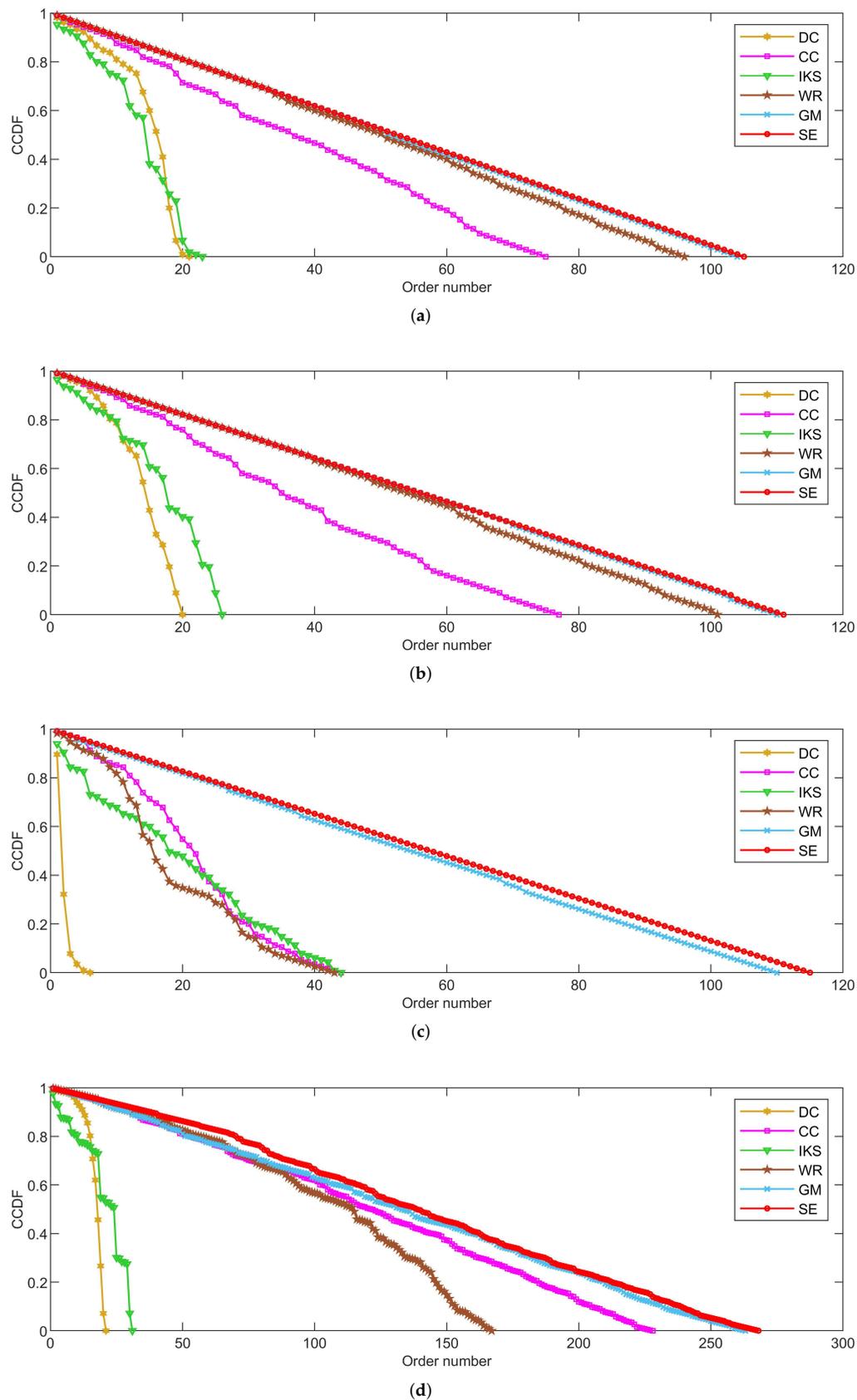


Figure 5. The curves of CCDF on (a) POLB; (b) ADJN; (c) FOOT; and (d) NETS datasets.

5.2.3. On EMAI and HAMS Datasets

Figure 6 is the curves of CCDF for two large-scale datasets, i.e., the EMAI and HAMS datasets. As can be seen, the performance of all methods can be divided into three categories roughly, i.e., well-performing SE and GM methods, moderately-performing CC and WR methods, and poorly-performing DC and IKS methods. Especially in Figure 6b, it should be pointed out that the curves of CCDF obtained by all methods have a process of vertical decline. The main reason for this phenomenon is that it is difficult to identify the importance of isolated nodes. Here, the importance score of an isolated node is set to the minimum in all methods. Although the importance of isolated nodes is not significant, this special structure can affect the importance scores of other nodes. From Figure 6b, it can be easily seen that the SE method still obtains the smoothest descending slope in all comparison methods. What is more, the biggest difference in total order numbers between SE and other methods can exceed 700. This further validates the effectiveness of SE method proposed in this paper.

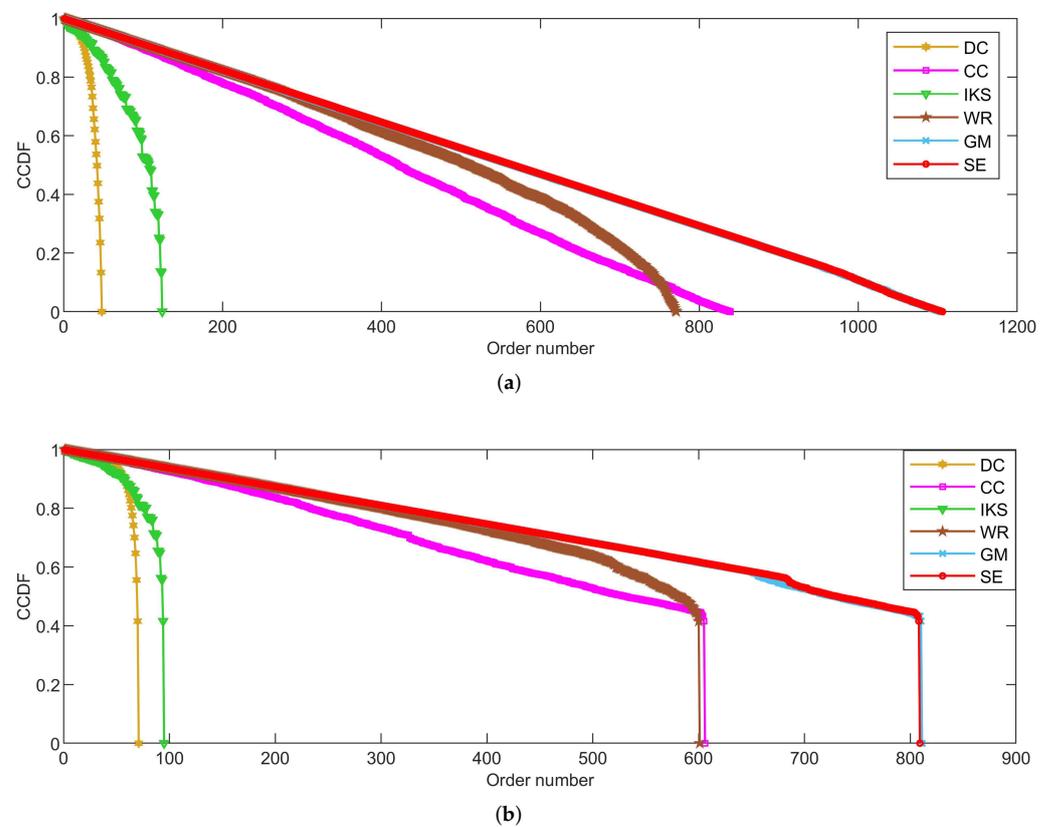


Figure 6. The curves of CCDF on (a) EMAI and (b) HAMS datasets.

5.3. Robustness Analysis

In this subsection, we evaluate the robustness of the SE method by comparing the curves of ζ and τ of SE method with that of other benchmark methods.

- The left side of Figures 7–10 is the curves of ζ , which shows the number changes of connected components. The horizontal axis of subfigures represents the proportion of removed nodes and the vertical axis represents the number of connected components after removing nodes from the dataset.
- The right side of Figures 7–10 is the curves of τ , which shows the variation of the maximum connected component. The horizontal axis of subfigures represents the proportion of removed nodes and the vertical axis represents the value of τ calculated by Equation (20).

The larger value of ζ and the smaller value of τ , the stronger the robustness of the corresponding ranking method. From the result, it can be found that the ζ curves of *SE* method obtain the faster uptrend and the τ curves of *SE* method obtain the faster downtrend in most datasets as the proportion of removed nodes increases.

5.3.1. On *CONT* and *LESM* Datasets

Figure 7 is the curves of ζ and τ on *CONT* and *LESM* datasets. As can be seen from Figure 7a, when the proportion of removed nodes changes from 10% to 90%, the *SE* method can always obtain the maximum value of ζ . Obviously, the *SE* method has the most obvious upward trend compared with other methods. In Figure 7b, when the proportion of removed nodes is only 10%, the τ value of all methods is equal to 0.9184 except *SE* method. In fact, the value of τ obtained by *SE* method only is 0.7959, which is 0.1225 lower than other methods. This advantage is more pronounced after the proportion of removed nodes reaches 30%.

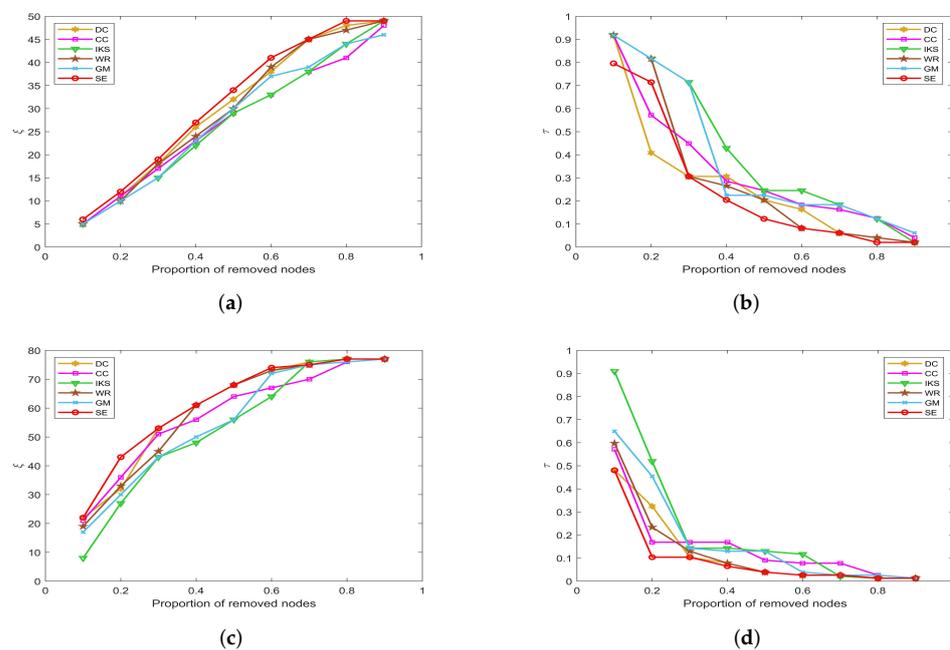


Figure 7. The curves of ζ and τ on (a,b) *CONT* and (c,d) *LESM* datasets.

On the *LESM* dataset, as shown in Figure 7c,d, the differences between the six methods are obvious. Especially when the proportion of removed nodes is 20%, the value of ζ corresponding to *DC*, *CC*, *IKS*, *WR*, *GM* and *SE* methods is 32, 36, 27, 33, 30 and 43, respectively. Obviously, the *SE* method is superior to other methods. What is more, the value of τ corresponding to the above methods is 0.3247, 0.1688, 0.5193, 0.2338, 0.4545 and 0.1039, respectively. One can find that the difference between the *SE* and *IKS* method is as high as 0.4154. That is to say, the maximum connected component of the *IKS* method contains 40 nodes, while that of the *SE* method contains only 8 nodes. This fully confirms that *SE* method has better robustness in small-scale datasets.

5.3.2. On *POLB*, *ADJN*, *FOOT* and *NETS* Datasets

Figure 8 is the curves of ζ and τ on *POLB* and *ADJN* datasets. With the increase in dataset scale, the robustness of the *CC* method decreases significantly, but the advantage of the *DC* method becomes more obvious. As shown in Figure 8a,c, both *DC* and *SE* methods obtain the same value of ζ in most cases. The main reason is that the *DC* method regards the nodes with larger degrees as the more important nodes, and these nodes can affect the number of connected components to a great extent. On the whole, the robustness of the *IKS* method is relatively poor. From Figure 8b,d, it can be found that the curves of τ

corresponding to the *SE* method can maintain the fastest downtrend when the proportion of removed nodes starts from 30%.

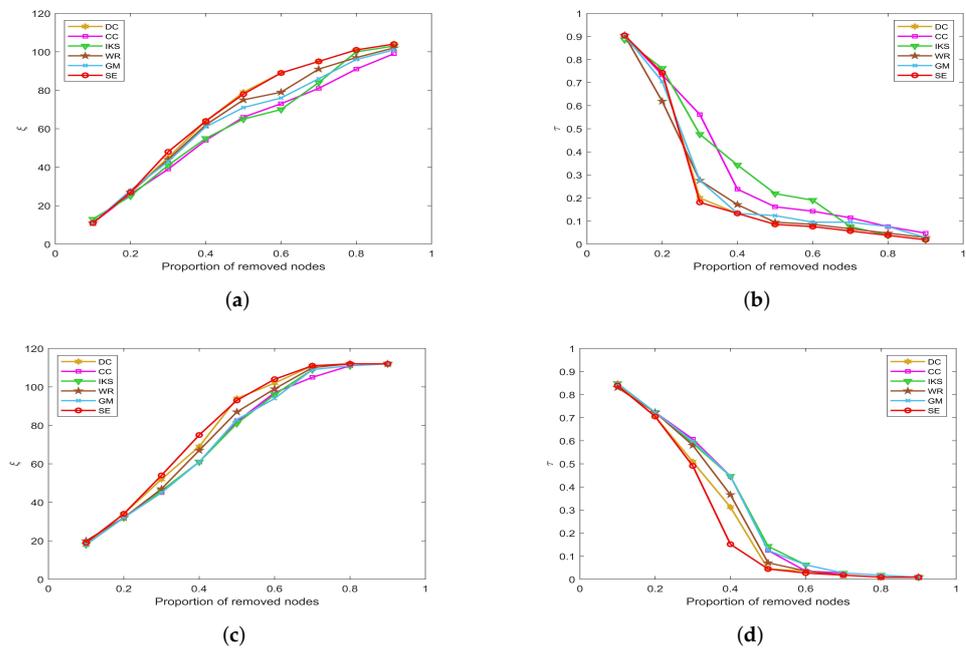


Figure 8. The curves of ζ and τ on (a,b) *POLB* and (c,d) *ADJN* datasets.

Figure 9 is the curves of ζ and τ on *FOOT* and *NETS* datasets. One can observe that all methods obtain similar ranking sequences on *FOOT* dataset. As shown in Figure 9a,b, six ranking methods show the same robustness until the proportion of removed nodes is as high as 50%. However, in fact, when the proportion of removed nodes is greater than 50%, the *SE* method obtains the largest value of ζ , and the *CC* method obtains the smallest value of τ .

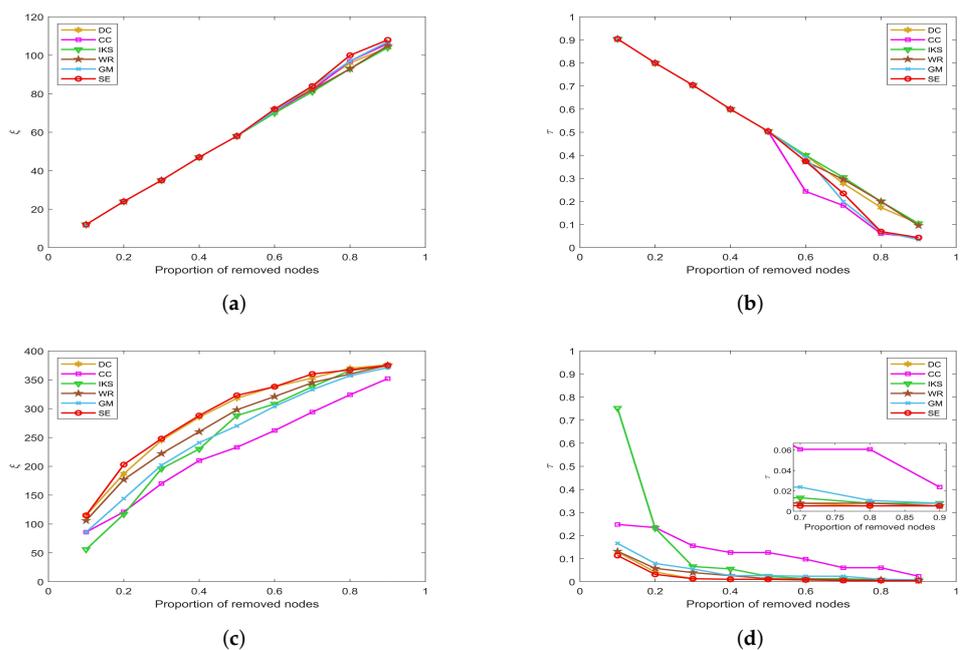


Figure 9. The curves of ζ and τ on (a,b) *FOOT* and (c,d) *NETS* datasets.

It is a pity that the *CC* method does not show better robustness on datasets with more nodes, such as the *NETS* dataset. From Figure 9c, it can be found that the value of ζ obtained

by the *CC* method is much lower than that of other methods, and the maximum difference between *CC* and *SE* methods reaches as high as 90. Similarly, as shown in Figure 9d, the value of τ corresponding to the *CC* method is much higher than other methods, and the maximum difference between *CC* and *SE* methods is as high as 0.2031. For this, we can guess that the *SE* method would show more excellent robustness on large-scale dataset.

5.3.3. On *EMAI* and *HAMS* Datasets

Figure 10 is the curves of ζ and τ on *EMAI* and *HAMS* datasets. As shown in Figure 10a,b, the *SE* and *DC* methods can maintain absolute superiority compared with other benchmark methods. Table 2 shows that the *EMAI* dataset has the largest value of m and $\langle d \rangle$ is as high as 9.6230. For this kind of tightly connected large-scale dataset, the *CC* and *IKS* methods perform poorly, and *WE* and *GM* methods are always in the middle position. The advantages of *DC* and *SE* methods are not easy to distinguish. It should be pointed out that all methods are close to the minimum value of τ when the proportion of removed nodes is greater than 40%. However, in fact, when the proportion of removed nodes is equal to 40%, the *SE* method is significantly better than other methods. This fully confirms that the *SE* method has better robustness compared with other benchmark methods.

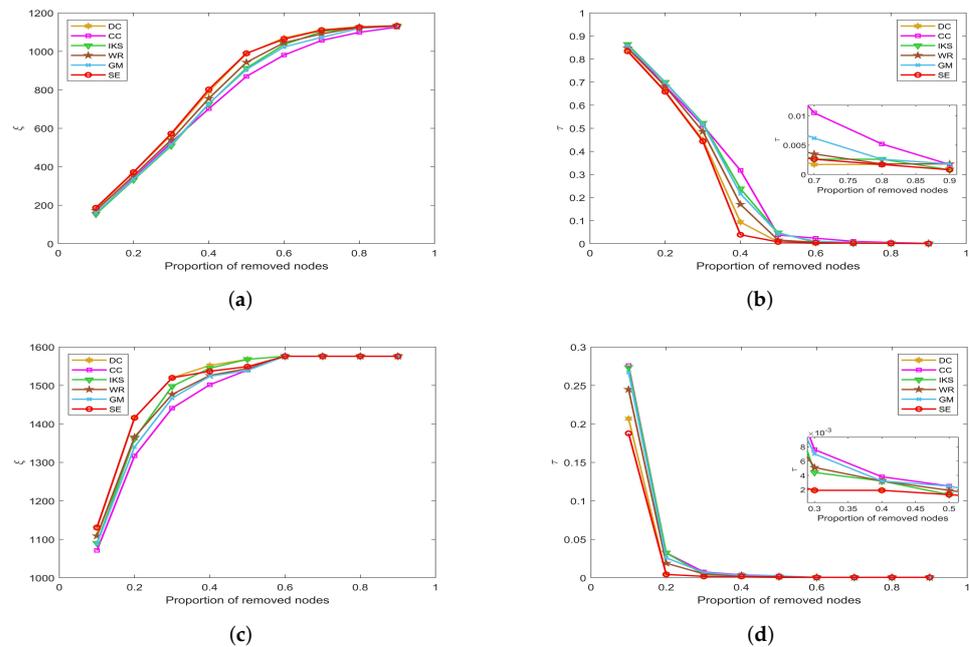


Figure 10. The curves of ζ and τ on (a,b) *EMAI* and (c,d) *HAMS* datasets.

By observing Figure 10c,d, the value of ζ and τ will not change after the proportion of removed nodes is greater than 60%. The reason is that the *HAMS* dataset contains 655 isolated nodes, which is more than 40% of the total number of nodes. However, in fact, the *SE* method can obtain the minimum value of τ when the proportion of removed nodes is between 10% and 50%. This further verifies that the *SE* method is also more robust for datasets with special structures.

5.4. Accuracy Analysis

In this part, we mainly analyze the accuracy of the *SE* method to identify key nodes in terms of the *SIR* model. Herein, we select the top 2, 4, 6, 8 and 10 nodes listed in the front of the ranking sequence as seeds for datasets with $n \leq 1000$. For datasets with more than 1000 nodes, we select top 20, 40, 60, 80 and 100 nodes as seeds. In terms of the *SIR* model, one can find that the disease cannot spread if the infected probability β is too small. The main reason is that the seeds have only a small probability to affect their neighbors.

Conversely, when the infected probability is too high, all nodes will become infected state. This is meaningless for accuracy analysis. Therefore, we mainly consider the propagation ability of seeds at the threshold of infected probability [46], i.e., $\beta = 1 / (\langle d \rangle - 1)$ and $\gamma = 1$.

Figures 11–13 show the propagation ability of seeds obtained by six ranking methods on eight real-world datasets. From the results, one can find that the SE method can obtain a more accurate ranking sequence.

5.4.1. On CONT and LESM Datasets

Figure 11 is the propagation ability of key nodes obtained by different methods on two small-scale datasets. Obviously, the SE method shows a more pronounced upward trend. That is to say, the top 10 key nodes obtained by the SE method have much higher propagation ability compared with other benchmark methods. Especially for the CONT dataset, the maximum propagation ability of SE method is 0.6361, which is 0.1495 higher than that of IKS method. Similarly, the maximum propagation ability of the SE method is 0.4963 on the LESM dataset, which is 0.0751 higher than that of the IKS method. Certainly, the IKS method performs poorly in most experiments. It can be seen from the previous experiments that the IKS method is not clear to identify the importance of different nodes. As a result, these nodes obtain the lowest propagation ability.

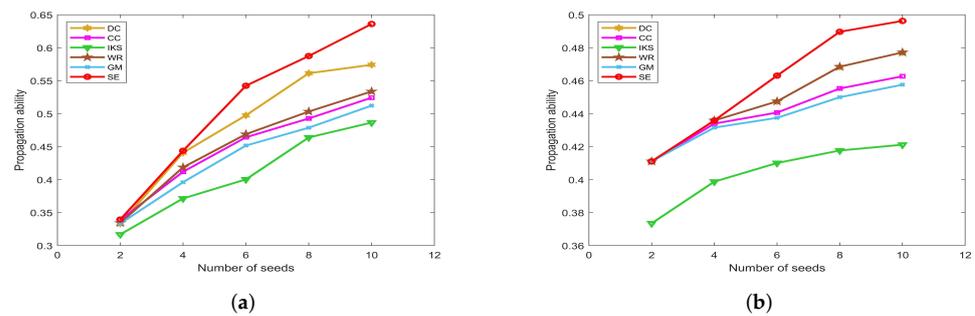


Figure 11. The propagation ability of seeds on (a) CONT and (b) LESM datasets.

Figure 11a shows the fact that the DC method is obviously superior to the WR method on the CONT dataset. However, the propagation ability curves of DC and WR methods completely coincide on LESM dataset, as shown in Figure 11b. This is mainly because the top 10 nodes obtained by these two methods are the same. As can be seen from the foregoing discussion, both DC and WR methods consider the degree information of nodes. If the dataset contains many nodes with the same degree, the accuracy of DC and WR methods will decrease significantly. On the contrary, the DC and WR methods can obtain more accurate ranking sequences for datasets with significantly different degrees of nodes, such as the LESM dataset. However, in fact, the SE method takes the local and global structure information into account, and the accuracy of it is obviously better than that of DC and WR methods.

5.4.2. On POLB, ADJN, FOOT and NETS Datasets

Figure 12 is the propagation ability of key nodes obtained by different methods on POLB, ADJN, FOOT and NETS datasets. On POLB dataset, the propagation ability curves of DC, WR, GM and SE methods all have an obvious upward trend, while the IKS is still the worst-performing method as shown in Figure 12a.

By observing Figure 12b,c, it can be found that the distribution of propagation ability curves is relatively dense. The main reason is that most of the methods obtain the same key nodes. For example, all methods treat nodes 18 and 3 as the top 2 key nodes except IKS method on ADJN dataset. Therefore, most methods achieve similar propagation ability curves. In this case, it should be pointed out that the SE method still has a slight advantage.

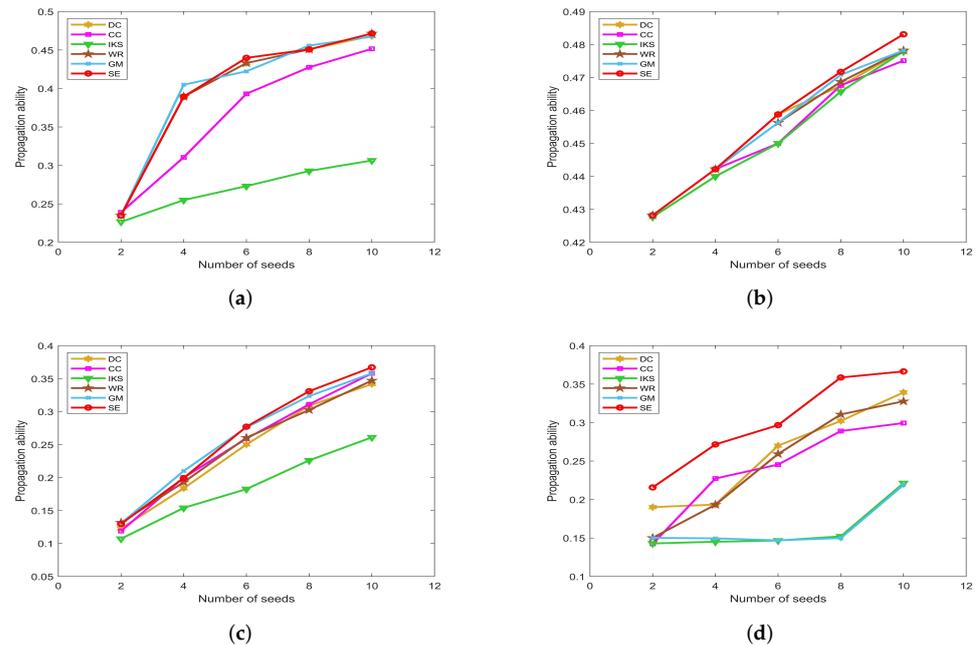


Figure 12. The propagation ability of seeds on (a) POLB; (b) ADJN; (c) FOOT; and (d) NETS datasets.

This advantage of the SE method is more significant on the NETS dataset. As can be seen from Figure 12d, it obtains the maximum propagation ability among all of the benchmark methods. Especially when the number of seeds is 8, the propagation ability of the SE method is 0.2087 higher than that of the GM method and 0.2065 higher than that of the IKS method. This means that the key nodes obtained by the SE method can infect 136 nodes, which is 79 higher than that of the GM method and 78 higher than that of the IKS method. Therefore, we can conclude that the ranking sequence obtained by the SE method is more accurate compared with other benchmark methods.

5.4.3. On EMAI and HAMS Datasets

Figure 13 is the curves of propagation ability on EMAI and HAMS datasets. As the scale of the dataset increases, the number of seeds we selected also increases to 100. From Figure 13a, one can find that the SE method can maintain the obvious upward trend. One can find that the SE method outperforms the other benchmark methods after the number of seeds exceeds 20. Since the special structure of the HAMS dataset, the variation range of propagation ability is small. For this, the SE method still has a slight advantage compared with other methods as shown in Figure 13b. This further confirms that the SE method has higher ranking accuracy compared with other benchmark methods.

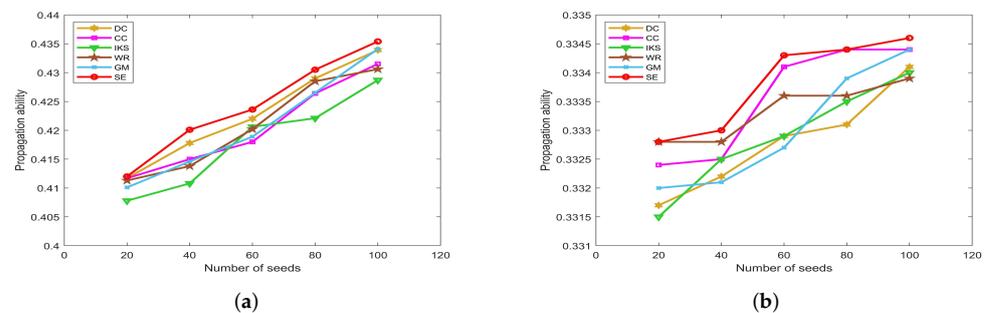


Figure 13. The propagation ability of seeds on (a) EMAI and (b) HAMS datasets.

5.5. Computational Complexity Analysis

Given that $G = (V, E)$ is a graph data with n nodes and m edges, and the proposed *SE* method includes two stages in the process of constructing the score function for nodes. Firstly, the computational complexity of calculating the local structure entropy is $O(sn)$, where s is the number of connected components in G . Secondly, the computational complexity of calculating the global structure entropy is $O(n)$. Therefore, the total computational complexity of *SE* method is $O(sn + n) = O(n(s + 1))$.

Table 4 lists the computational complexity of the proposed *SE* method and other benchmark methods. One can find that the computational complexity of the *CC* method is $O(nm)$, and that of the *GM* method is $O(n^2)$ [47]. Due to s being the number of connected components after removing the target node, the value of s is far smaller than m and n . That is to say, the computational complexity of the *SE* method is much lower than that of the *CC* and *GM* methods. Although *DC* and *IKS* methods have the lowest computational complexity, their performance is far worse than that of other methods in previous experiments. In general, although the computational complexity of the *SE* method is in the middle position among all comparison methods, it can obtain better ranking results.

Table 4. The computational complexity of six ranking methods.

Method	<i>DC</i>	<i>CC</i>	<i>IKS</i>	<i>WR</i>	<i>GM</i>	<i>SE</i>
Complexity	$O(n)$	$O(nm)$	$O(n)$	$O(m + n < d >)$	$O(n^2)$	$O(n(s + 1))$

6. Conclusions

In order to further explore the influence of structure information on node importance, this paper has designed a structure entropy-based node importance ranking method. The score function of node importance is constructed from the perspective of node removal, which transformed the importance of nodes into the global structure entropy of graph data. After removing the target node, the local structural entropy of the connected component is calculated by using the degree information of nodes. Furthermore, the global structure entropy of graph data is constructed in terms of the number of connected components. A large number of experiments demonstrated that the proposed method is more advantageous in aspects of monotonicity, node distribution and ranking accuracy.

Although the proposed method has better performance on most datasets, it is not hard to see that this paper only discussed the undirected and unweighted graph data with less than 2000 nodes due to the limitation of the experimental platform. In our following studies, we will seek more resources to verify the performance of the proposed method on larger-scale graph data and other types of graph data.

Author Contributions: Conceptualization, S.L. and H.G.; Writing—original draft, S.L. and H.G.; Methodology, S.L. and H.G.; Supervision, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (No.61966039), the Xingdian Talent Support Program for Young Talents (No.XDYC-QNRC-2022-0518) and the Scientific Research Fund Project of Education Department of Yunnan Province (No.2023Y0565).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We are hugely grateful to the possible anonymous reviewers for their constructive comments with respect to the original manuscript. At the same time, we acknowledge all the network data used in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Omar, Y.M.; Plapper, P. A survey of information entropy metrics for complex networks. *Entropy* **2020**, *22*, 1417. [[CrossRef](#)] [[PubMed](#)]
2. Liu, J.; Li, X.; Dong, J. A survey on network node ranking algorithms: Representative methods, extensions, and applications. *Sci. China Technol. Sci.* **2021**, *64*, 451–461. [[CrossRef](#)]
3. Wang, Z.; Du, C.; Fan, J.; Xing, Y. Ranking influential nodes in social networks based on node position and neighborhood. *Neurocomputing* **2017**, *260*, 466–477. [[CrossRef](#)]
4. Fei, L.; Deng, Y. A new method to identify influential nodes based on relative entropy. *Chaos Solitons Fractals* **2017**, *104*, 257–267. [[CrossRef](#)]
5. PastorSatorras, R.; Castellano, C.; Van Mieghem, P.; Vespignani, A. Epidemic processes in complex networks. *Rev. Mod. Phys.* **2015**, *87*, 925. [[CrossRef](#)]
6. Wang, W.; Tang, M.; Stanley, H.E.; Braunstein, L.A. Unification of theoretical approaches for epidemic spreading on complex networks. *Rep. Prog. Phys.* **2017**, *80*, 036603. [[CrossRef](#)]
7. Cui, A.; Wang, W.; Tang, M.; Fu, Y.; Liang, X.; Do, Y. Efficient allocation of heterogeneous response times in information spreading process. *Chaos Interdiscip. J. Nonlinear Sci.* **2014**, *24*, 033113. [[CrossRef](#)]
8. Davis, J.T.; Perra, N.; Zhang, Q.; Moreno, Y.; Vespignani, A. Phase transitions in information spreading on structured populations. *Nat. Phys.* **2020**, *16*, 590–596. [[CrossRef](#)]
9. Javier, B.H.; Yamir, M. Absence of influential spreaders in rumor dynamics. *Phys. Rev. E* **2012**, *85*, 026116.
10. Yao, X.; Gu, Y.; Gu, C.; Huang, H. Fast controlling of rumors with limited cost in social networks. *Comput. Commun.* **2022**, *182*, 41–51. [[CrossRef](#)]
11. Solá, L.; Romance, M.; Criado, R.; Flores, J.; García del Amo, A.; Boccaletti, S. Eigenvector centrality of nodes in multiplex networks. *Chaos: Interdiscip. J. Nonlinear Sci.* **2013**, *23*, 033131. [[CrossRef](#)]
12. Wen, T.; Deng, Y. Identification of influencers in complex networks by local information dimensionality. *Inf. Sci.* **2020**, *512*, 549–562. [[CrossRef](#)]
13. Zareie, A.; Sheikahmadi, A.; Jalili, M. Influential node ranking in social networks based on neighborhood diversity. *Future Gener. Comput. Syst.* **2019**, *94*, 120–129. [[CrossRef](#)]
14. Lu, P.; Zhang, Z.; Guo, Y.; Chen, Y. A novel centrality measure for identifying influential nodes based on minimum weighted degree decomposition. *Int. J. Mod. Phys. B* **2021**, *35*, 2150251. [[CrossRef](#)]
15. Chen, D.; Sun, H.; Tang, Q.; Tian, S.; Xie, M. Identifying influential spreaders in complex networks by propagation probability dynamics. *Chaos Interdiscip. J. Nonlinear Sci.* **2019**, *29*, 033120. [[CrossRef](#)]
16. Katz, L. A new status index derived from sociometric analysis. *Psychometrika* **1953**, *18*, 39–43. [[CrossRef](#)]
17. Li, J.; Yin, C.; Wang, H.; Wang, J.; Zhao, N. Mining Algorithm of Relatively Important Nodes Based on Edge Importance Greedy Strategy. *Appl. Sci.* **2022**, *12*, 6099. [[CrossRef](#)]
18. Wang, M.; Li, W.; Guo, Y.; Peng, X.; Li, Y. Identifying influential spreaders in complex networks based on improved k-shell method. *Phys. A Stat. Mech. Its Appl.* **2020**, *554*, 124229. [[CrossRef](#)]
19. Yang, Y.; Hu, M.; Huang, T. Influential nodes identification in complex networks based on global and local information. *Chin. Phys. B* **2020**, *29*, 088903. [[CrossRef](#)]
20. Zareie, A.; Sheikahmadi, A. A hierarchical approach for influential node ranking in complex social networks. *Expert Syst. Appl.* **2018**, *93*, 200–211. [[CrossRef](#)]
21. Freeman, L.C. A set of measures of centrality based on betweenness. *Sociometry* **1977**, *40*, 35–41. [[CrossRef](#)]
22. Goldstein, R.; Vitevitch, M.S. The influence of closeness centrality on lexical processing. *Front. Psychol.* **2017**, *8*, 1683. [[CrossRef](#)] [[PubMed](#)]
23. Yang, X.; Xiao, F. An improved gravity model to identify influential nodes in complex networks based on k-shell method. *Knowl.-Based Syst.* **2021**, *227*, 107198. [[CrossRef](#)]
24. Yang, P.; Liu, X.; Xu, G. A dynamic weighted TOPSIS method for identifying influential nodes in complex networks. *Mod. Phys. Lett. B* **2018**, *32*, 1850216. [[CrossRef](#)]
25. Maurya, S.K.; Liu, X.; Murata, T. Graph neural networks for fast node ranking approximation. *ACM Trans. Knowl. Discov. Data* **2021**, *15*, 78. [[CrossRef](#)]
26. Liu, C.; Cao, T.; Zhou, L. Learning to rank complex network node based on the self-supervised graph convolution model. *Knowl.-Based Syst.* **2022**, *251*, 109220. [[CrossRef](#)]
27. Gray, R.M. *Entropy and Information Theory*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.
28. Ma, L.; Ma, C.; Zhang, H.; Wang, B. Identifying influential spreaders in complex networks based on gravity formula. *Phys. A Stat. Mech. Its Appl.* **2016**, *451*, 205–212. [[CrossRef](#)]
29. Bernadette, B.M.; Christophe, M. Entropy and monotonicity in artificial intelligence. *Int. J. Approx. Reason.* **2020**, *124*, 111–122.
30. Fan, W.; Liu, Z.; Hu, P. Identifying node importance based on information entropy in complex networks. *Phys. Scr.* **2013**, *88*, 065201.
31. Zareie, A.; Sheikahmadi, A.; Fatemi, A. Influential nodes ranking in complex networks: An entropy-based approach. *Chaos Solitons Fractals* **2017**, *104*, 485–494. [[CrossRef](#)]

32. Guo, C.; Yang, L.; Chen, X.; Chen, D.; Gao, H.; Ma, J. Influential nodes identification in complex networks via information entropy. *Entropy* **2020**, *22*, 242. [[CrossRef](#)]
33. Zhong, L.; Bai, Y.; Tian, Y.; Luo, C.; Huang, J.; Pan, W. Information entropy based on propagation feature of node for identifying the influential nodes. *Complexity* **2021**, *2021*, 5554322. [[CrossRef](#)]
34. Yu, Y.; Zhou, B.; Chen, L.; Gao, T.; Liu, J. Identifying Important Nodes in Complex Networks Based on Node Propagation Entropy. *Entropy* **2022**, *24*, 275. [[CrossRef](#)]
35. Zhang, Q.; Li, M.; Deng, Y. A new structure entropy of complex networks based on nonextensive statistical mechanics. *Int. J. Mod. Phys. C* **2016**, *27*, 1650118. [[CrossRef](#)]
36. Lei, M.; Cheong, K.H. Node influence ranking in complex networks: A local structure entropy approach. *Chaos Solitons Fractals* **2022**, *160*, 112136. [[CrossRef](#)]
37. Ai, X. Node importance ranking of complex networks with entropy variation. *Entropy* **2017**, *19*, 303. [[CrossRef](#)]
38. Liu, Y.; Liu, S.; Yu, F.; Yang, X. Link prediction algorithm based on the initial information contribution of nodes. *Inf. Sci.* **2022**, *608*, 1591–1616. [[CrossRef](#)]
39. He, W.; Liu, S.; Xu, W.; Yu, F.; Li, W.; Li, F. On rough set based fuzzy clustering for graph data. *Int. J. Mach. Learn. Cybern.* **2022**, *13*, 3463–3490. [[CrossRef](#)]
40. Fu, Y.H.; Huang, C.Y.; Sun, C.T. Using global diversity and local topology features to identify influential network spreaders. *Phys. A Stat. Mech. Its Appl.* **2015**, *433*, 344–355. [[CrossRef](#)]
41. Lu, P.; Zhang, Z. Critical nodes identification in complex networks via similarity coefficient. *Mod. Phys. Lett. B* **2022**, *36*, 2150620. [[CrossRef](#)]
42. Li, Y.; Cai, W.; Li, Y.; Du, X. Key node ranking in complex networks: A novel entropy and mutual information-based approach. *Entropy* **2019**, *22*, 52. [[CrossRef](#)]
43. Chen, X.; Zhou, J.; Liao, Z.; Liu, S.; Zhang, Y. A novel method to rank influential nodes in complex networks based on tsallis entropy. *Entropy* **2020**, *22*, 848. [[CrossRef](#)]
44. Li, P.; Wang, S.; Chen, G.; Bao, C.; Yan, G. Identifying Key Nodes in Complex Networks Based on Local Structural Entropy and Clustering Coefficient. *Math. Probl. Eng.* **2022**, *2022*, 8928765. [[CrossRef](#)]
45. Kudryashov, N.A.; Chmykhov, M.A.; Vigdorowitsch, M. Analytical features of the SIR model and their applications to COVID-19. *Appl. Math. Model.* **2021**, *90*, 466–473. [[CrossRef](#)]
46. Sheng, J.; Zhu, J.; Wang, Y.; Wang, B.; Hou, Z. Identifying influential nodes of complex networks based on trust-value. *Algorithms* **2020**, *13*, 280. [[CrossRef](#)]
47. Ullah, A.; Wang, B.; Sheng, J.; Long, J.; Khan, N.; Sun, Z. Identification of nodes influence based on global structure model in complex networks. *Sci. Rep.* **2021**, *11*, 6173. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.