

Research Article

Identification of Influential Nodes via Effective Distance-based Centrality Mechanism in Complex Networks

Aman Ullah ¹, Bin wang ¹, Jinfang Sheng ¹, Jun Long ^{1,2} and Nasrullah Khan ^{3,4}

¹School of Computer Science and Engineering, Central South University, Changsha, 410083, China

²Big Data Institute, Central South University, Changsha 410083, China

³College of Computer Science and Technology, Nanjing University of Aeronautic and Astronautic, Nanjing 210016, China

⁴Department of Computer Science, COMSATS University Islamabad, Vehari Campus, Vehari 61100, Pakistan

Correspondence should be addressed to Jinfang Sheng; jfsheng@csu.edu.cn

Received 13 July 2020; Revised 7 September 2020; Accepted 25 January 2021; Published 9 February 2021

Academic Editor: Lucia Valentina Gambuzza

Copyright © 2021 Aman Ullah et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Efficient identification of influential nodes is one of the essential aspects in the field of complex networks, which has excellent theoretical and practical significance in the real world. A valuable number of approaches have been developed and deployed in these areas where just a few have used centrality measures along with their concerning deficiencies and limitations in their studies. Therefore, to resolve these challenging issues, we propose a novel effective distance-based centrality (EDBC) algorithm for the identification of influential nodes in concerning networks. EDBC algorithm comprises factors such as the power of K-shell, degree nodes, effective distance, and numerous levels of neighbor's influence or neighborhood potential. The performance of the proposed algorithm is evaluated on nine real-world networks, where a susceptible infected recovered (SIR) epidemic model is employed to examine the spreading dynamics of each node. Simulation results demonstrate that the proposed algorithm outperforms the existing techniques such as eigenvector, betweenness, closeness centralities, hyperlink-induced topic search, H-index, K-shell, page rank, profit leader, and gravity over a valuable margin.

1. Introduction

In recent years, complex networks are an attractive and hot research area by virtue of its wide range of practical and theoretical applications in many major fields [1–5]. Several real-world application areas such as management science, chemistry, economics, and financial systems [6, 7], computer science, biological science [8, 9], and many other similar fields can be regarded as complex networks [10, 11]. Finding the most important nodes is helpful to efficiently analyze the entire network, such as controlling the spreading of disease, detecting the most vital node of the disease transmission rapidly can control the spreading of disease, and erecting a new marketing tool [12–15]. Some of the existing commonly used approaches to identify the influential nodes are closeness centrality (CC) [16], degree centrality (DC) [17], betweenness centrality (BC) [18], information centrality (IC) [19], load centrality (LC) [20], eigenvector centrality (EC) [21], page rank (PR) [22], H-index

[23], K-shell decomposition, hyperlink-induced topic search (HITS) [24, 25], and so on. Several algorithms are usually not precise, but precise algorithms have comparatively high computational time complexity. For example, DC is a very simple method of finding influential nodes, but it only considers limited information. Therefore, it is not suitable in some cases. Similarly, BC and CC are not applicable in large networks because of their high computation time complexity. PR is based on global information, so it works very well in directed networks but is not suitable for undirected ones [26]. Similarly, the K-shell algorithm takes into account the core or periphery position of the nodes in the networks to investigate influential nodes. It is a straightforward index mechanism for finding the influential nodes; due to its low complexity, it can be applied in large-scale networks, although it cannot differentiate among the key nodes in the same core layer [27–29]. All in all, the above existing approaches have low accuracy of ranking and are ineffective for some networks. There are numerous ranking

approaches for influential or key nodes identification, but it is not easy to find all influential nodes. Frequently, two types of models are used to evaluate ranking results of important nodes: (i) SIR model [30] and (ii) Kendall's correlation coefficient τ [31]. The main job of the SIR model is to choose the top-k ranking node's results as seed nodes. When the selected seed nodes or vertices of a technique make the flow of network spread faster, then the technique is said to be better than the other techniques. For measurement between two ranking results, Kendall's correlation coefficient τ is used. If Kendall's correlation coefficient τ is positive, then it means the two rankings are positively correlated, and if Kendall's correlation coefficient τ is negative, then the two rankings are negatively correlated. If the coefficient is 1, it means the rankings are the same, and if it is -1, it means the rankings are completely opposite. The technique is said to be more accurate when Kendall's correlation coefficient τ is higher. From the above discussion, we propose a new, efficient, and effective method termed as EDBC algorithm for key or influential nodes identification in complex networks. EDBC comprises effective distance, K-shell, nodes degree, and many other levels of neighbor influences or neighborhood potential-like factors. We applied EDBC on nine different real-world networks. The simulation results proved that our algorithm is better than other conventional algorithms, including newly developed algorithms such as gravity centrality and profit leader, which were, respectively, developed in 2017 and 2018.

The core contribution of the proposed EDBC algorithm is summarized as follows. (1) A new ranking centrality perspective: from the last two decades, several models and algorithms regarding the identification of influential or key nodes have been developed but still it is a challenge. In this regard, we propose a novel effective distance-based centrality algorithm which is comprising of several features to have experimented on unweighted networks and structure to sort out the important nodes. (2) Accuracy: we have compared the proposed algorithm with various ranking methods where experiments have shown that EDBC is comparatively more effective and efficient algorithm. Through rendering comprehensive experiments on nine real-world networks, the EDBC has outperformed various existing algorithms such as BC, CC, EC, PR, HITS, H-index, K-shell, GR, and PL. (3) Parameter-free: the proposed EDBC neither depends on prior knowledge nor relies on adjustments of parameters. It has its own way to quickly and efficiently identifying the key nodes. (4) Applicability: the proposed EDBC reduces the calculation cost as its basis on two-stage neighbor nodes. Therefore, its application is appropriate on any type of networks, i.e., directed or undirected networks.

The rest of this paper is organized as follows. In Section 2, we present overviews of the related work. In Section 3, we discuss the proposed EDBC algorithm, whereas its performance evolution is discussed in Section 4. Finally, we elaborate conclusion and future recommendations in Section 5.

2. Related Work

The criterion of detecting the most important nodes in complex networks is one of the challenging issues from the past few decades. Two types of centrality measures can be

used for identification of key nodes, i.e., (i) matrix-based centrality such as random walk matrix [32], Laplacian matrix [33], and adjacency matrix [34] (ii) superficial-based properties such as CC, BC, DC, and motif centrality measurement [35]. Currently, a vast number of studies have been offered regarding the identification of the key nodes, e.g., node removals, profit leader [36], and local neighbor contribution [3]; these studies have their own advantages and disadvantages. Zhong-Kui Bao et al. [37] proposed heuristic clustering (HC) method of detecting the most influential nodes; HC works on the basis of similarity index which categorizes nodes into various clusters; in this way, the center nodes in clusters are taken as multiple spreaders. Hongming Mo et al. [38] proposed an evidential method for the influential node identification, which is based on the Dempster-Shafer evidence theory. Furthermore, Bian and Yong [39] introduced a new evidential centrality (NEC) algorithm, which is the extension of the evidential method. Tian et al. [40] proposed an analytic hierarchy process (AHP), which works on the basis of multiple attribute decision-making (MADM) method; AHP is used to detect the important branch of every decision and choose the best nodes in the entire network. Zeng et al. [41] have proposed a mixed degree decomposition algorithm for finding ranking nodes; however, the limitation of this technique is the high time complexity and high degree peripheral nodes. Lin et al. [42] improve the K-shell decomposition method; they consider the shortest distance among the nodes and set of nodes by high index K-shell. For further improvement of the K-shell decomposition method, Bae et al. [43] suggested neighborhood coreness centrality (cn) and extended neighborhood coreness (cn+) algorithms. In [44], authors proposed a centrality method for influential nodes identification, which is based on gravity formula. In this method, mass is replaced by the K-shell value of each node and the distance is the shortest path between two nodes. According to the authors, the proposed gravity-based method has better results than other existing methods. Liu et al. [45] proposed a weight degree centrality measure method, which considered the degree of nodes, neighborhoods of their nodes, and one tuning parameter. They compared the proposed WDC algorithm with other various algorithms such as BC, DC, K-shell decomposition, neighborhood coreness, and extended neighborhood coreness methods. A neighborhood centrality method was proposed by Liu et al. [46] for influential nodes identification, which cogitates numerous neighbors' level influence factors that are based on benchmark centrality measures. On the contrary, there were some indexing centrality methods which are based on edges' potential to enhance the effectiveness of ranking centrality measures [47, 48]. Pei et al. [49] have claimed that the K-shell decomposition algorithm performs well for finding key nodes in the entire real network. In [50], a bio-inspired centrality method was proposed, which combines the K-shell index, with Physarum centrality to identify key nodes. Gomez et al. [51] improve the BC centrality method to take into account explicitly numerous dimensions. Furthermore, Zang et al. [52] proposed an advanced unbiased betweenness centrality method, which considers a reverse

propagation algorithm to choose the key nodes in the real network. There have been numerous studies to improve the existing algorithm for influential nodes identification. Though, all these studies have their own advantages and disadvantages. Still, efficient and effective influential nodes identification persists as a nontrivial challenge. Inspired from [44, 53], we proposed an efficient algorithm called EDBC, which effectively identifies the highly influential nodes in various scales of complex networks.

3. EDBC Model

In this section, we discuss some basic concepts about the proposed method. The proposed method considers the degree of nodes, effective distance, power of K-shell, and several levels of neighborhood nodes influences. EDBC calculation is presented in Sections 3.1 and 3.2.

3.1. Preliminaries

Definition 1. (degree $d(v_i)$). In graph theory, the degree of a node denotes the number of all connected edges in the node, that is, the sum of all connected edges in the network. In graph G , set $A = \{A_{ij} | i, j = 1 \dots N\}$ denotes the adjacency matrix of G which is defined as follows:

$$d(v_i) = \sum_{j=1}^N a_{ij}, \quad (1)$$

where represents node degree, v_i , and a_{ij} indicates the element of A .

Definition 2. (power of K-shell $\alpha(i, j)$). We consider the nodes i and j for K-shell (k_{si} and k_{sj}). We add k_{si} and k_{sj} in equation (2) for the measurement of power influences:

$$\alpha(i, j) = \sqrt{k_{si} + k_{sj}}, \quad (2)$$

where k_{si} and k_{sj} represent the K-shell of node i and node j . Here, we used the square root for normalization of the influence factor for the power of K-shell.

Definition 3. (effective distance $ED(i, j)$, see [54]). Effective distance is an essential parameter in the networks $G = (N, M)$ spreading process, and we employ this parameter in the EDBC algorithm. For calculating the distance between the nodes, we have not used the existing distance path algorithm such as Dijkstra and the Bellman–Ford algorithm because the time complexities of these algorithms are very high. For this purpose, we propose effective distance between the nodes, which can be calculated through the following formula:

$$ED(i, j) = (1 - \log P_{ji}) \geq 1, \quad (3)$$

where P_{ji} denotes the flow of information ratio from node V_j to V_i : $P_{ji} = (a_{ji} / \sum_{i \in \Psi(j)} a(ji))$. As can be seen from the formula, $ED(i, j) \neq ED(j, i)$.

Definition 4. (influence $(I_{i,j})$). Normally, when the neighbors of nodes have high influence, then nodes' influences will increase automatically. Besides, the influence of nodes on neighbors will be decreased with an increase of shortest distances among them. Inspired from inverse square law, we can compute the influence of node as follows:

$$(I_{i,j}) = \sum_{j \in \eta(i)} \frac{\alpha(i, j) + d(v_i)}{ED(i, j)^2}. \quad (4)$$

The sum of node influence on all neighbor nodes, which measure the influence of that node, which can be calculated as follows:

$$EDBC(i) = \sum_{j \in \eta(i)} (I_{i,j}), \quad (5)$$

where $\eta(i)$ is the set of nearest all neighbors node i .

Algorithm 1 depicts the working mechanism of our proposed solution.

3.2. EDBC Algorithm. The node influence in a network depends on three aspects. (i) The node location in the network when a node is in the center of the network, and the influence will be high. Otherwise, if the node is located on the edge of the network, the influence will definitely be relatively small. (ii) The number of node neighbors: a large number of neighbors will be the greater influence of the node. (iii) The distance between the nodes: the smaller the distance is, there is a greater probability of information transmission between nodes and its neighbors. On this basis, we proposed EDBC algorithm, as described in section 1. The proposed EDBC algorithm mainly comprises factors such as the power of K-shell, degree nodes, effective distance, and numerous levels of neighbor influence. At the first stage, it calculates the degree and K-shell of nodes in network, and then, it finds the power of K-shell according to equation (2). Furthermore, it calculates the distance between the nodes (nearest and next-nearest neighbors) through equation (3), and finally, it calculates the influence of each node using equations (4) and (5). To further demonstrate the EDBC algorithm over a specific calculation process, we consider a simple network to provide a clear and detailed picture of the proposed methodology. Figure 1 presents a synthetic network with 16 nodes and 21 edges, where node V16 has a higher influence than other nodes in the network. Moreover, a control flowchart of EDBC algorithm is also presented through Figure 2, where, in the first phase, we have constructed a concerning network; then, we calculate the degree and K-shell of each node; after that, we calculate the power of K-shell as well as effective distance that only considers the nearest and the next-nearest neighbors nodes. At the last step, influence of each node in the entire network is hereby calculated.

3.2.1. Computation of the Power of K-Shell $\alpha(i, j)$. For calculating the power of K-shell, here we apply equation (2). For node V16, the K-shell value is 2; first of all, we calculate the K-shell of node i and node j . Table 1 shows the K-shell of

```

Input:  $G = (N, M)$ ;
(1) for each node  $v$  in  $G = (N, M)$  do
(2)   calculate the degree of  $v$ 
(3)   calculate the K-shell of  $v$ 
(4)   //Computing the interaction in the most nearest neighbors nodes
(5)   for node  $u$  in  $G$ .neighbors of node  $v$  do
(6)     evaluate the power of K-shell ( $v$ ) using equation (2)
(7)     evaluate the power of K-shell ( $u$ ) using equation (2)
(8)     evaluate  $ED(v,u)$  using equation (4)
(9)     evaluate  $influence(v,u)$  using equation (5)
(10)  //Computing the interaction in the next-nearest neighbors' nodes
(11)  for node  $w$  in  $G$ .neighbors of node  $u$  do
(12)    evaluate the power of K-shell ( $v$ ) using equation (2)
(13)    evaluate the power of K-shell ( $w$ ) using equation (2)
(14)    evaluate  $ED(v,w)$  using equation (3)
(15)    evaluate  $influence(v,w)$  using equation (4)
(16)  end for
(17) end for
(18) end for
(19) //Calculate influence  $I(v)$  of all neighbours' nodes
(20) evaluate  $EDBC(v)$  using equation (5)
(21) return  $EDBC(v)$ 
Output: Ranked nodes

```

ALGORITHM 1: EDBC.

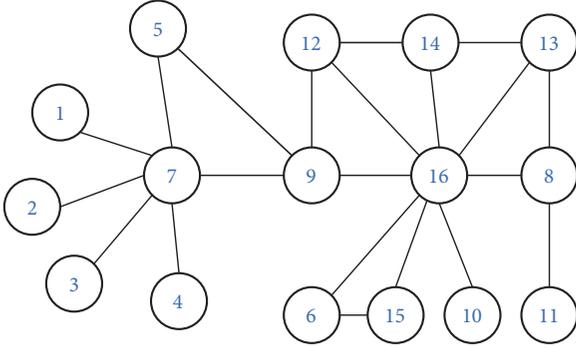


FIGURE 1: A simple network with 16 nodes and 21 edges.

given toy network. We have $\alpha(16, 9) = \sqrt{(2+2)} = 2$, $\alpha(16, 7) = \sqrt{(2+2)} = 2$, and $\alpha(16, 2) = \sqrt{(1+2)} = 1.7320$. Similarly, in this way, other K-shell power can be calculated.

3.2.2. Calculation of the Effective Distance. ED (between nearest and next-nearest neighbor of nodes) depends on the closeness among the nodes and its neighbors, to calculate effective distance among the neighbors and the next neighbors of V16 by using equation (3):

$$ED(16, 9) = (1 - \log P_{9,16}) = 1.903, \quad (6)$$

where

$$P_{9,16} = \frac{1}{d(16)} = 0.125. \quad (7)$$

Now, to calculate the distance between the next neighbors of V16, the distance can be

$$ED(16, 7) = ED(16, 9) + ED(7, 9),$$

$$ED(16, 7) = 1.903 + (1 - \log P_{9,7}) = 1.903 + 1.7783 = 3.6813. \quad (8)$$

Similarly,

$$ED(16, 5) = ED(16, 9) + (1 - \log P_{9,5}) = 1.903 + 1.301 = 3.204. \quad (9)$$

3.2.3. Calculation of the Neighbor Nodes' Influence. In this context, EDBC algorithm considers only the nearest and the next-nearest neighbor's nodes. Here, we take V9 as an example of the nearest neighbor node, according to equation (4):

$$I_{(16,9)} = \frac{\alpha(16, 9) + d(16)}{ED(16, 9)^2} = 2.7614, \quad (10)$$

where $d(16)$ is the degree of V16, which is 8. Now, we take the next-nearest neighbor of V9. Therefore, the influence of node V16 to node V7 or node V5 can be computed as follows:

$$I_{(16,7)} = \frac{\alpha(16, 7) + d(16)}{ED(16, 7)^2} = 0.7379. \quad (11)$$

By this rule, all the nearest and next-nearest neighbor's nodes of the influence with V16 can be calculated.

3.2.4. Computation of the Influence for Each Node. The computation of the total influence of the nodes in the entire network means the sum of the total number of influence

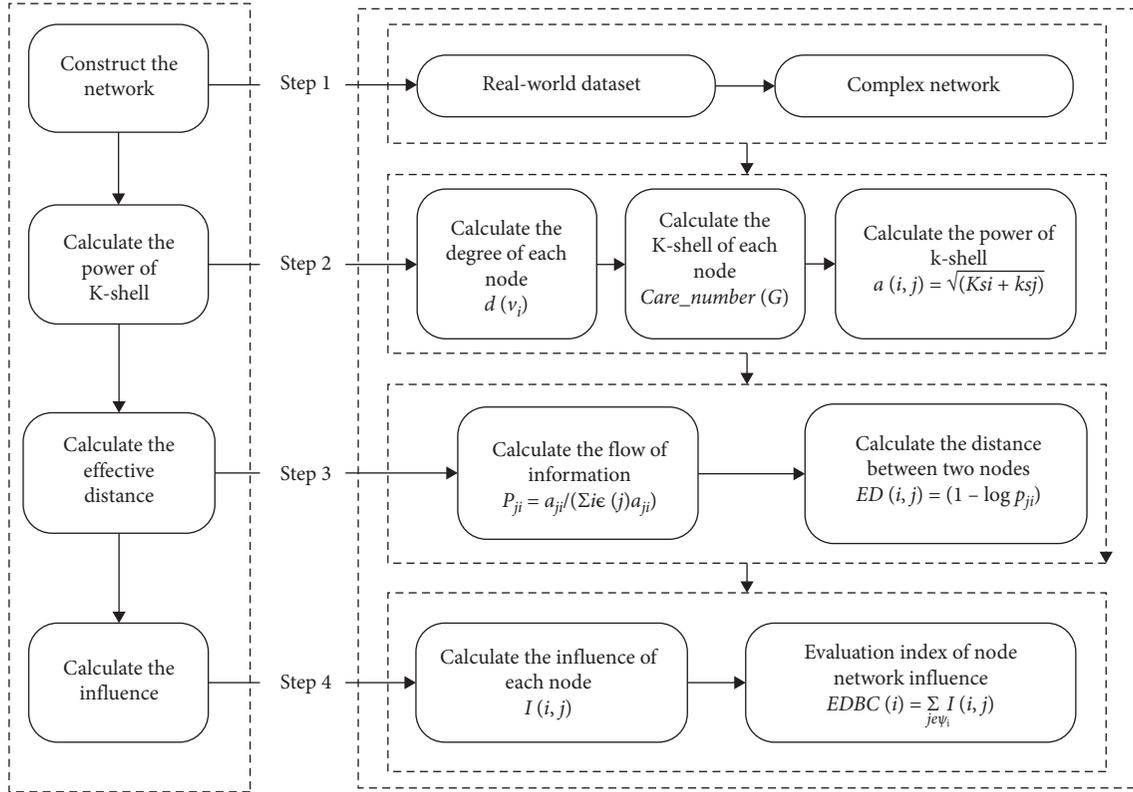


FIGURE 2: The flowchart of the proposed EDBC.

TABLE 1: K-shell and degrees of each node in a toy network.

Node	1	7	2	3	4	5	9	6	16	15	8	11	13	12	10	14
Degree	1	6	1	1	1	2	4	2	8	2	3	1	3	3	1	3
Core number/K-shell	1	2	1	1	1	2	2	2	2	2	2	1	2	2	1	2

nodes on the nearest and next-nearest neighbor nodes. By using equation (5), we can compute the total influence of the node V16 for a given graph:

$$EDBC(V16) = \sum_{j \in \Psi} I_{(16,j)} + \sum_{k \in (i), k \neq 1} I_{(16,k)} = 57.6251, \quad (12)$$

where j and k show the nearest and next-nearest neighbor nodes. Applying all of the above procedure, we get all nodes influence the ranking results of the given toy network, as shown in Table 2.

3.3. Computational Complexity. EDBC takes three components. In the first phase, it calculates the degree and K-shell, so the time complexity becomes $O(|M|)$. In the second phase, it computes the effective distance among the nodes and their adjacent or neighbors. The distance between the nearest and next-nearest neighbors in the entire network is calculated. Therefore, the computational time complexity is $O(N < K >^2)$, and in the third and fourth phases, the node's influence of all neighbors will be calculated. Therefore, EDBC algorithm's total time complexity can become $O(N < k >^2 + |M|)$, where N and k denote the number of

nodes and average degree of nodes in the network, respectively.

4. Experimentation and Results' Analysis

In this section, we discuss the experimental setup we performed on various real-world networks and datasets to overview of the performance evaluation and comparison between the proposed EDBC algorithm and ten aforementioned state-of-the-art algorithms. We implemented EDBC algorithm in *Python 3.7* and rendered experiments on Feiteng Server (1, 16-core, and 1.5gHZ) provided by Kirin Operating System in the main lab of our school.

4.1. Comparison of Some Benchmark Centralities. We compared EDBC of our proposed methodology with nine baseline algorithms, which we are going to briefly summarize and classify according to their characteristic in this section. The existing centrality measures are classified into categories:

- (a) Structure-based centralities: the node influence is significantly affected via the network topology. In

TABLE 2: The influence ranking results using EDBC of the given toy network.

Node	16	7	9	12	8	14	13	6	15	5	10	1
Influence	57.6	45.9	37.4	27.2	26.4	25.2	24.7	18.9	18.9	17.8	15.9	10.7

fact, a lot of centralities consider structural information for the identification of influential nodes. The structure-based measure is further divided into two categories: (i) based on the neighborhood of each node, for example, K-shell and H-index and (ii) based on paths between nodes, such as betweenness and closeness centralities. K-shell (KS) algorithm decomposes a network into sublayers that are directly connected to the centrality [55]. In fact, this algorithm assigns Ks (integer index) to each node, which are representatives of the nodes' location in the network. The Ks of each node with high or low values is placed at the center or periphery of the network. By this rule, the network is designated via a layered structure, revealing the complete hierarchy of its nodes. The innermost nodes are known as core or nucleus, and K-shell nodes are placed into internal and external layers. It is a straightforward indexing algorithm for identifying the influential nodes and can be applied to any network; however, it cannot differentiate the influential nodes in the same core layer. Therefore, KS algorithm is not suitable for some of the networks. Our experiments provide evidence in this case. The H-index [23] algorithm is commonly used to evaluate researchers' and scientists' academic achievements. The computation process of the H value in the H-index signifies that the author or scholar has at least H citations of H article published by an author. High H-index reflects that the node has greater influence. Still, this algorithm has many problems with its effective execution, e.g., edge value weight needs to be in appropriate range, otherwise the desired ranking could not be achieved. Betweenness centrality (BC) [18] is used to compute important nodes via global information. It works on the shortest path between the nodes where a node with a higher BC value indicates that it is more important than other nodes in the entire network. However, it is not appropriate for large and complex networks, whereas experimental results have clearly depicted that EDBC is a suitable approach for any kind of network whether it is large or small. Closeness centrality (CC) [16] depends on global information, and it works on the relative distance between each pair of nodes to detect the important nodes. CC can better be identifying influential nodes but very difficult to apply in large networks. The limitation of this algorithm is the lack of applicability to networks by disconnected components. In short, these kinds of centralities have high computational complexity and cannot be suitable for large-scale networks.

- (b) Eigenvector-based centralities: eigenvector-based centralities not only consider the neighborhood

node number but also consider their influences. These centralities are PageRank, eigenvector, profit leader, HITS, and gravity index. Eigenvector centrality (EC) [21] is based on the information gain method to choose important nodes in the network. It neither depends only on the degree number of neighborhood nodes nor the impact of each neighbor node. This algorithm has excessive applications both in theoretical and practical. Though, if there exist several nodes with higher degrees in the graph, the risky phenomenon of fractional convergence will still occur; therefore, EC is not an appropriate solution for such networks. PageRank (PR) [22] is one of the famous centrality algorithms; Google search engine is working based on this algorithm. Like EC, PR supposes that the influence of a web page depends on both the quantity and quality of the pages joined by it [55]. It is suitable for directed networks but not for undirected and unweighted, but EDBC is appropriate in case of any type of network whether it is directed or undirected. hyperlink-induced topic search (HITS) [25] is a link analysis technique that uses various metrics concurrently. It scrutinizes the influence of the node via two attributes, i.e., hub and authority value, where the hub value emulates the node role for information transmission, while the authority value analyzes the original node creativity in the information. These two attributes interact and converge through the iteration process. Profit leader (PL) [36] is one of the most recent algorithms, which was proposed by Yu et al. in May 2018. This algorithm is based on the profit leader concept, which chooses the important nodes in the entire network by calculating the profit the node can make. This algorithm is very simple and applicable to some networks. However, this algorithm is not working very well in case of small networks. Gravity (GR) was proposed by Ma et al. in 2016 [44], which works on the principle of gravity formula. Here, mass is replaced by the K-shell value of each node, and the distance is the shortest path between two nodes, to identify the key nodes in the network. Its drawback is the computational time complexity which is too high to be suitable for large-scale networks.

4.2. Data Description. We evaluated EDBC on nine real-world networks to reveal its performance. The data has been collected from different genres of fields where its basic properties, scales, and structures are presented in Table 3. These datasets are publicly available and can be downloaded from sites <http://konect.uni-koblenz.de/networks/>, <http://snap.stanford.edu/data/as-caida.html>, and <http://networkrepository.com/web-spam.php>.

TABLE 3: The basic statistics of nine real-world networks, where $|M|$, $|N|$, and $\langle K \rangle$ represent the number of edges, number of nodes, and average degree, respectively. The last two columns indicate the maximum degree (Kmax) and cluster coefficients (CC) of the networks.

Networks	$ M $	$ N $	$\langle K \rangle$	$\langle K \text{ max} \rangle$	$\langle CC \rangle$
Physicians	1098	241	9.112	34	0.31157
e-mail	5451	1333	9.615	71	0.2202
Subelj-euroroad	1417	1174	2.410	5	0.01673
Air-traffic control	2615	1226	4.2659	37	0.04
P-friendships	12534	1858	5.745	85	0.167
US-powergrid	6594	4941	2.67	10	0.0801
Web-spam	37375	4767	15.681	477	0.2859
PGP algorithm	24316	10680	4.5536	205	0.2659
CAIDA-project	53381	26475	4.0326	2628	0.208

Physicians [56]. This dataset contains 241 nodes and 1098 edges, collected in 1966 by RonBurt. A node denotes a physician, and the link between two physicians represents that the left-hand physician is the friend of the right-hand physician. There is only one link that subsists between any two nodes.

e-mail [57]. This network is an e-mail system at the University RV in south Tarragona, Spain. It consists of 5451 links and 1133 nodes. Nodes denote users, and links show that one e-mail was sent at least.

Subelj-euroroad [58]. This is a Europe e-road network, which contains 1174 nodes and 1417 edges. Nodes represent cities, and edges of the two nodes indicate that they are linked via an e-road.

Air-traffic control [59]. This network was created for FAA (federal aviation administration) system in the USA. Nodes represent service centers or airports, and edges are formed from a string of preferred routes.

Petster-friendships [60]. This network consists of friendships among users on the website hamsterster.com. In this network, users represent nodes, and the closeness among users represents edges.

US-Powergrid [61]. This is an undirected network which consists of information about the power grid of the Western states in the USA. It contains 6594 links and 4941 nodes. The generator or transformer in the network represents nodes, while the power supply line denotes edges.

Web-spam [62]. This dataset is constructed by Purdue University network repository. It consists of 4767 nodes and 37,375 edges. Pages and hyperlinks represent nodes and edges, respectively.

PGP algorithm [63]. This is the user's communication network for the Pretty Good Privacy (PGP) algorithm and consists of 24,316 links and 10,680 nodes. It comprises only the giant interconnected elements of the network.

CAIDA-project [64]. This is the CAIDA-project network, collected in 2007 and containing 26,475 nodes

and 53,381 edges. A node represents autonomous systems (AS), while edges represent communication.

4.3. Evaluation Metrics

4.3.1. SIR Model. To evaluate the effectiveness of the identification of influential nodes, we employ the SIR model to simulate the spreading influence of ranking nodes in our experiments [65, 66]. This model consists of three states, i.e., S (susceptible) means a healthy state and may be infected by others. I (infected) means an infected state and can infect other individuals. R (recovered) represents a recovered state, which cannot be infected by other individuals again. All the seed nodes in the network are in a susceptible state initially. At each time step, the seed node in the network may infect its neighbor's susceptible nodes via a probability β ; then, infected nodes are recovered (enter into the recovered state) with probability μ . This process will continue until there has no longer infected nodes. Finally, all the recovered nodes are used to imitate the real node influence.

$S(t)$, $I(t)$, and $R(t)$ denote the nodes' numbers in susceptible, infected, and recovered states. So,

$$\begin{cases} \frac{ds(t)}{dt} = -\beta s(t)i(t), \\ \frac{di(t)}{dt} = \beta s(t)i(t) - \beta i(t), \\ \frac{dr(t)}{dt} = \beta i(t). \end{cases} \quad (13)$$

The spreading influence K_i of the node V_i is as follows:

$$K_i = F(t) = \frac{1}{N_{\text{ite}}} (nI + nR), \quad (14)$$

where N_{ite} denotes iteration numbers; here, we set $N_{\text{ite}} = 1000$ independent run, and nI and nR represent the number of infected and recovered nodes, respectively.

4.3.2. Kendall's Correlation Coefficient τ . Kendall's correlation coefficient τ [67, 68] is used to measure the correctness results of two ranking methods. In this paper, we acquired Kendall's correlation coefficient to measure the performance of the proposed algorithm. Let us suppose that two node sequences (X & Y) are correlated with similar nodes' number (n), $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_n)$. One pair of two annotations (x_i, y_i) and (x_j, y_j) ($i \neq j$) are said to be concordant if the ranking of both component agree, i.e., if both $x_i > x_j$ and $y_i > y_j$ or $x_i < x_j$ and $y_i < y_j$. They are said to be discordant if $x_i > x_j$ and $y_i < y_j$ or $x_i < x_j$ and $y_i > y_j$ or if $x_i = x_j$ or $y_i = y_j$, the pair is neither concordant nor discordant. Kendall's τ is defined as follows:

$$\tau(X, Y) = \frac{n_c - n_d}{0.5n(n-1)}, \quad (15)$$

TABLE 4: Comparison of a toy network’s ranking results of EDBC with other various algorithms, where GR, PL, and PR represent profit leader, page rank, and gravity, respectively. Here, EDBC* and SIR* indicate the ranking value of the EDBC and SIR value, respectively.

BC	CC	EC	H-index	HITS	K-shell	PR	PL	GR	EDB	EDBC*	SIR	SIR*
16	16	16	9	16	16	16	16	16	16	57.63	16	1.93
9	9	12	12	12	12	7	7	9	7	45.78	7	1.62
7	7	14	13	14	13	9	9	7	9	37.38	9	1.60
8	12	9	14	9	14	8	8	12	12	27.24	8	1.41
12	5	13	16	13	9	13	14	13	8	26.37	14	1.49
13	8	8	6	8	6	12	13	14	14	25.17	13	1.44
14	13	6	7	15	7	14	12	8	13	24.65	12	1.51
1	14	15	8	6	8	5	15	6	6	18.96	15	1.31
2	6	7	15	7	15	6	6	15	15	18.96	6	1.32
3	15	10	5	10	5	15	5	5	5	17.83	5	1.34
4	10	5	11	5	11	1	11	10	10	15.89	11	1.13
5	1	11	2	11	2	2	4	2	1	10.70	4	1.17
6	2	1	3	1	3	3	1	3	2	10.70	1	1.15
15	3	2	4	2	4	4	2	4	3	10.70	2	1.16
11	4	3	1	3	1	11	3	11	4	10.70	3	1.15
10	11	4	10	4	10	10	10	1	11	06.69	10	1.17

where n_c and n_d represent the number of concordant and discordant pairs, respectively. If Kendall’s correlation coefficient (t) is positive, then the two rankings are positively correlated, and if Kendall’s correlation coefficient (t) is negative, then two rankings are negatively correlated. The coefficient is 1 when the two rankings are the same, and if the coefficient is -1, then it means the two rankings are completely opposite. Higher Kendall’s values specify more precise and better performance.

4.4. Performance Evaluation. In this experiment, we used the SIR and Kendall’s models to verify the effectiveness of EDBC. First, we have used a toy network as an example, as shown in Figure 1, where we applied EDBC to identify the influential nodes, compared it with nine baseline algorithms, and listed the ranking results and SIR values of each node, as shown in Table 4. It can be seen that the EDBC performs well than other various ranking algorithms in terms of identification of influential nodes’ ranking. Figure 3 illustrates Kendall’s correlation coefficient (τ) calculation results of the ten algorithms, where the ranking effects are generated via the K-shell, HITS, H-index, PageRank, BC, CC, EC, PL, and GR.

In Figure 3, EDBC gets the highest Kendall’s τ correlation coefficients in (physicians, e-mail, Subelij-eroroad, petster-friendships, air-traffic control, US-powergrid, and web-spam networks. The results in the range between 0.8 and 1 such as 0.9 for physicians and web-spam, 0.95 for e-mail, Subelij-eroroad, CAIDA-project, and US-powergrid, and 1.0 for PGP algorithm and petster-friendships, respectively. Figure 3 shows the performance between ten algorithms based on the comparison of the Kendall’s τ , and EDBC performs well on all kinds of networks. We note that the performance of BC measure is always the worst on all networks because BC is usually based on the definition of the shortest paths between the nodes and information in most networks which does not flow along with the shortest paths [69].

In order to further examine the performance measures of EDBC. We used the SIR model for spreading the impact of

the ranked nodes, where we used a relatively small β for big networks such as PGP algorithm and CAIDA-project. Especially, we have set $\beta = 0.01$ due to the bigger value, and the propagation will occur across the whole network [70]; in this situation, it will be hard to separate the influence of different nodes. For small networks such as web-spam, US-powergrid, physicians, e-mail, Subelij-eroroad, petster-friendships, and air-traffic control, we have set $\beta = 0.1$ to estimate the influences of each node. Therefore, either we achieve the propagation efficiency in the entire network for all nodes or we keep the recovery rate probability, i.e., γ is equal to 1 and the time t is equal to 1000.

The ranking results of the average number of infected nodes by the ten algorithms’ comparison are shown in Figure 4. A more influential node can affect more nodes; therefore, an efficient algorithm can produce a curve that reduces from left to right showing that EDBC is comparatively capable of the best performance on physicians, e-mail, Subelij-eroroad, air-traffic control, US-powergrid, and web-spam networks. Performance-wise, on petster-friendships network, the GR and EDBC algorithms are approximately the same, but still, EDBC works well in other networks; it is why all the curves are drawn smoothly with a negligible variation. For CAIDA-Project, all the algorithms show similar effects, but EDBC still provides a greater spreading effect than other algorithms. Moreover, ranking results of the performance of ten mentioned algorithms on the Subelij-eroroad network are presented in Table 5, where each node is considered a seed node that has recursively infected its neighbor nodes. It can be seen that EDBC performs better than other baseline algorithms. In addition, we performed the influence comparison of the top-10 nodes, which are distinctly selected by EDBC and other various algorithms. All top-10 distinct nodes are selected as seed nodes, and the time is set to t where values’ range of t varies from 1 to 20. As shown in Table 6, EDBC has highest propagation capability for all nodes, and it clearly shows that when the infection $F(t)$ increases as time t increase, we finally get a steady value at a time t after having consecutive

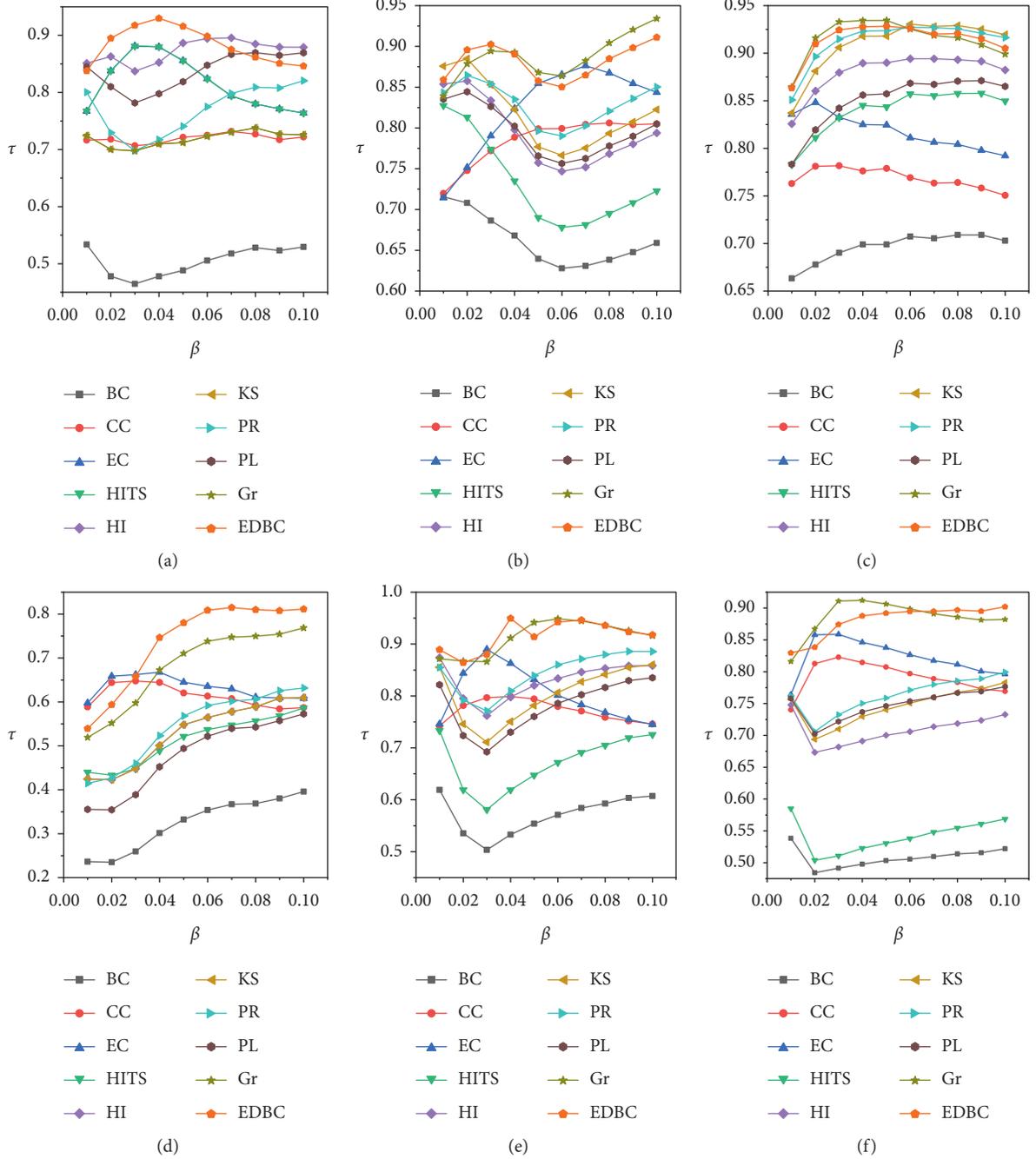


FIGURE 3: Continued.

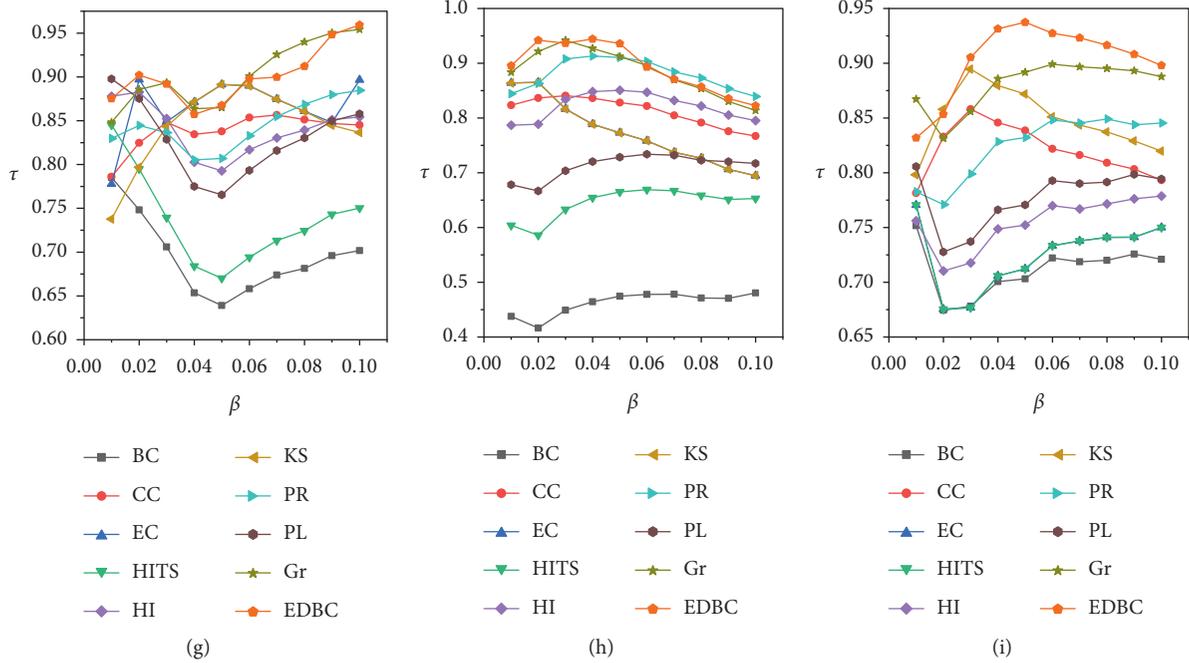


FIGURE 3: Results of Kendall's τ obtained from mutual comparison of results generated by ten mentioned algorithms and ranking of results produced of the SIR model. Network's spreading probability β was set in the range from 0.01 to 0.10, and results were calculated based on average outcome of 1000 independent runs. (a) Physicians. (b) e-mail. (c) Subelij-euroroad. (d) Air-traffic control. (e) Petster-friendships. (f) Web-Spam. (g) US-Powergrid. (h) PGP algorithm. (i) CAIDA-project, etc.

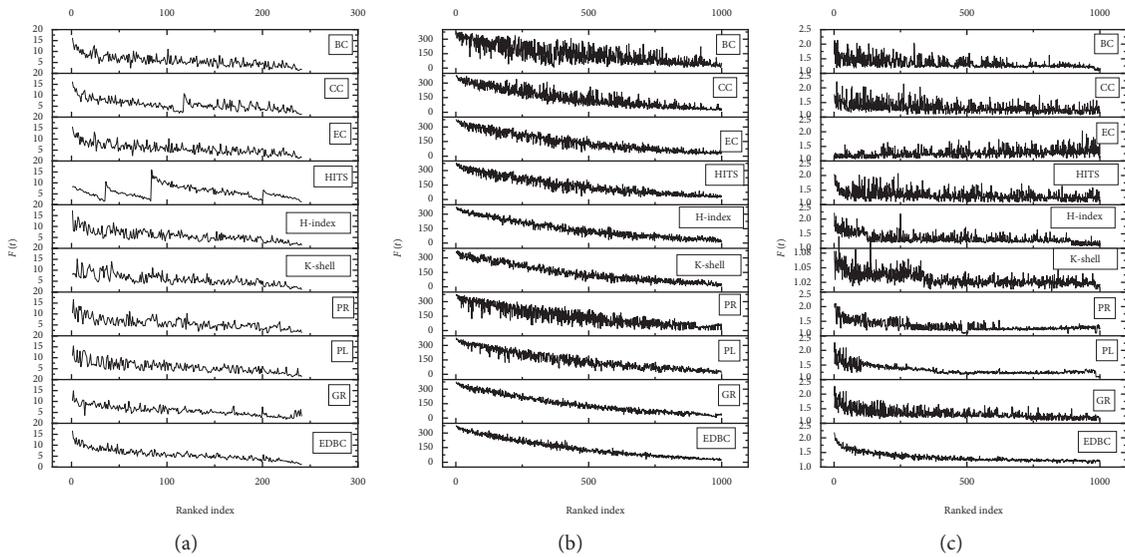


FIGURE 4: Continued.

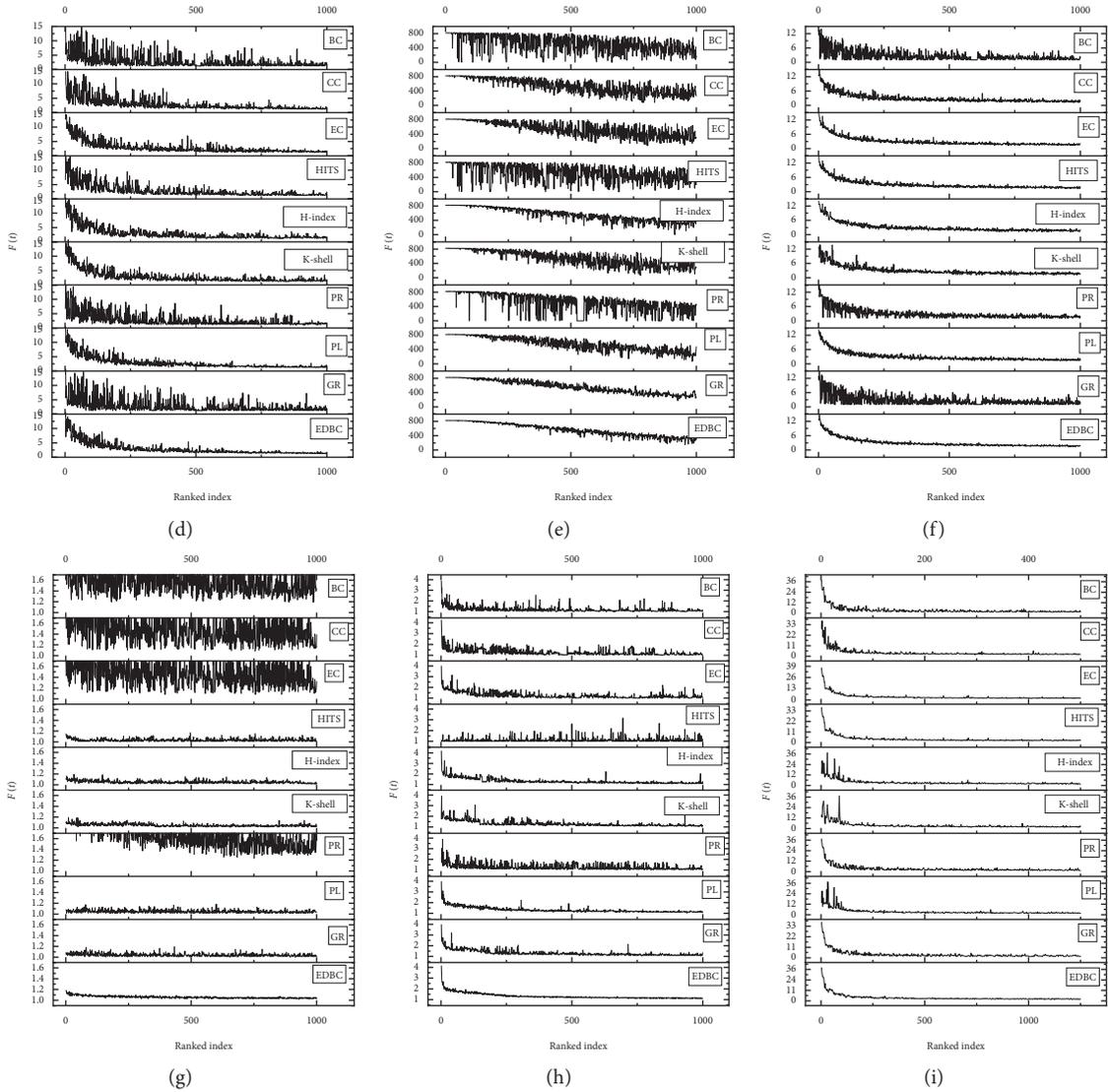


FIGURE 4: Influence of propagation for ranking effect using ten various algorithms, where $F(t)$ represents the number of infected and recovered nodes at time (t), and horizontal index line represents the ranking order list. (a) Physicians. (b) e-mail. (c) Subelj-euroroad. (d) Air-traffic control. (e) Petster-friendships. (f) Web-Spam. (g) US-Powergrid. (h) PGP algorithm. (i) CAIDA-project, etc.

TABLE 5: Ranking results of the SIR model w.r.t node influence of ten mentioned algorithms in Subelj-euroroad network. Here, GR, PL, and PR represent profit leader, page rank, and gravity, respectively.

t	BC	CC	EC	H-index	HITS	K-shell	PR	PL	GR	EDB
1	1.672	1.77	1.23	2.04	2.21	1.08	2.09	1.99	1.95	2.18
2	2.13	1.60	1.12	1.97	1.75	1.06	1.99	2.17	2.27	2.12
3	1.63	1.50	1.11	1.75	1.76	1.06	1.97	2.02	1.53	2.07
4	1.73	1.49	1.28	1.66	1.93	1.03	2.00	1.96	1.79	1.91
5	1.34	1.46	1.09	2.01	1.91	1.05	2.06	1.91	1.94	1.96
6	1.56	1.72	1.23	1.75	1.58	1.06	2.11	1.93	1.96	1.99
7	2.03	1.33	1.20	1.77	2.10	1.08	1.74	1.94	2.03	1.99
8	1.37	1.31	1.25	1.69	1.81	1.06	1.51	1.95	1.49	1.98
9	1.27	1.58	1.24	1.63	1.71	1.07	1.64	1.85	1.71	1.98
10	1.18	2.21	1.09	1.81	1.81	1.04	1.40	1.89	1.72	1.93
11	1.27	1.77	1.20	1.62	1.76	1.05	1.99	1.97	1.87	1.80
12	1.48	1.35	1.22	1.74	1.65	1.06	1.58	1.90	1.53	1.90
13	2.13	1.84	1.23	1.45	1.53	1.05	2.01	1.75	1.49	2.03
14	1.41	1.47	1.23	1.55	1.86	1.06	1.87	1.79	1.48	1.88

TABLE 5: Continued.

t	BC	CC	EC	H-index	HITS	K-shell	PR	PL	GR	EDB
15	1.28	1.44	1.14	1.54	1.50	1.07	1.69	1.76	1.59	1.82
16	1.39	1.27	1.17	1.52	1.65	1.04	1.97	1.72	1.36	1.83
17	1.35	1.51	1.12	1.77	1.70	1.04	1.77	1.80	1.98	1.82
18	1.73	1.56	1.09	1.72	1.52	1.11	1.63	1.74	1.82	1.74
19	1.33	1.56	1.09	1.49	1.69	1.08	1.79	1.81	1.94	1.87
20	1.24	1.63	1.09	1.35	1.78	1.07	1.99	1.76	1.58	1.89

TABLE 6: Influences $F(t)$ of top-10 distinct nodes using EDBC and other various algorithms in the e-mail network. Here, GR, PL, and PR represent profit leader, page rank, and gravity, respectively.

t	BC	CC	EC	H-index	HITS	K-shell	PR	PL	GR	EDBC
1	26.57	26.36	25.05	22.83	26.19	24.36	27.55	25.92	26.49	26.98
2	36.39	35.99	33.91	30.58	36.94	32.51	36.47	35.95	36.75	38.14
3	41.53	40.92	39.18	35.14	41.77	36.87	42.87	41.35	41.83	44.30
4	44.62	43.64	40.88	38.29	45.42	39.34	45.31	44.03	45.59	46.70
5	45.16	45.89	41.67	39.44	46.32	40.67	46.78	45.25	46.95	48.24
6	46.74	46.56	42.75	39.91	47.81	42.07	47.84	45.69	47.10	49.27
7	47.51	47.78	43.59	39.98	47.95	42.67	48.12	47.06	47.61	49.86
8	47.53	46.89	43.13	41.10	47.75	41.89	48.63	47.64	47.21	49.69
9	47.11	46.34	43.09	41.06	47.60	41.91	48.41	48.40	48.88	48.49
10	47.26	46.81	44.48	41.79	48.24	41.75	48.16	47.42	48.68	48.69
11	47.71	47.26	44.32	41.90	47.35	42.70	48.70	46.95	48.25	49.14
12	46.18	46.89	44.04	39.77	47.16	42.51	48.19	47.77	47.68	48.15
13	46.73	47.77	43.36	39.89	47.55	42.15	49.39	47.48	47.19	49.11
14	47.16	47.80	43.99	40.73	47.85	42.29	48.30	46.67	48.70	49.19
15	47.10	47.75	44.22	40.52	47.42	42.55	49.38	47.53	48.36	49.32
16	47.33	47.49	42.96	40.53	48.68	42.33	48.95	47.53	47.78	50.63
17	48.43	45.68	43.76	39.93	47.54	43.03	48.59	46.69	48.43	50.12
18	48.41	46.89	43.67	40.45	48.62	42.03	49.07	48.30	48.85	49.12
19	48.06	47.01	44.00	39.29	47.34	41.74	48.92	47.17	48.49	48.89
20	47.51	46.81	44.17	40.43	48.29	42.46	48.40	47.47	48.52	50.016

TABLE 7: Comparison of ranking of the top-10 nodes; owing to the limited space, only the top-10 nodes of the PGP algorithm are presented, where PR, PL, and GR denote page rang, profit leader, and gravity, respectively.

BC	CC	EC	H-index	HITS	K-shell	PR	PL	GR	EDBC
1251	1251	1251	1251	2562	4	960	1251	1251	338
1474	338	1533	26	136	1251	880	1408	26	1251
338	1474	1540	1533	2645	1266	194	1533	1533	26
1802	1282	1408	1540	3071	1300	338	1312	1474	1474
960	1802	1537	1408	3088	1301	41	1540	1408	1312
1034	1312	1530	1530	1577	1408	1251	26	1540	1272
253	1034	26	1266	1531	1530	588	338	1312	1548
250	1621	1935	1537	943	1533	1474	1537	1548	31
1338	250	1548	1300	3765	1537	250	1530	1530	1547
761	1338	1943	1548	4758	1540	372	1474	1537	1275

time points. Table 7 shows the top-10 ranked nodes. Due to the limited space, we have only shown the top-10 nodes of PGP algorithm network. Consequently, there are ten seed nodes, and most network propagation arrives at a steady-state at time period $t=20$ where we can examine the spreading effect of EDBC and all some other baseline algorithms. Moreover, Figure 5 indicates that the EDBC

algorithm has a good spreading efficiency of top-10 nodes. Specifically, EDBC has better performance on the physicians, e-mail, Subelij-euroroad, air-traffic control, petster-friendships, and PGP algorithm US-Powergrid networks. In petster-friendships and CAIDA-project networks, PR and EC have the best propagation effect. And, EDBC has also better spreading efficiency than other baseline algorithms.

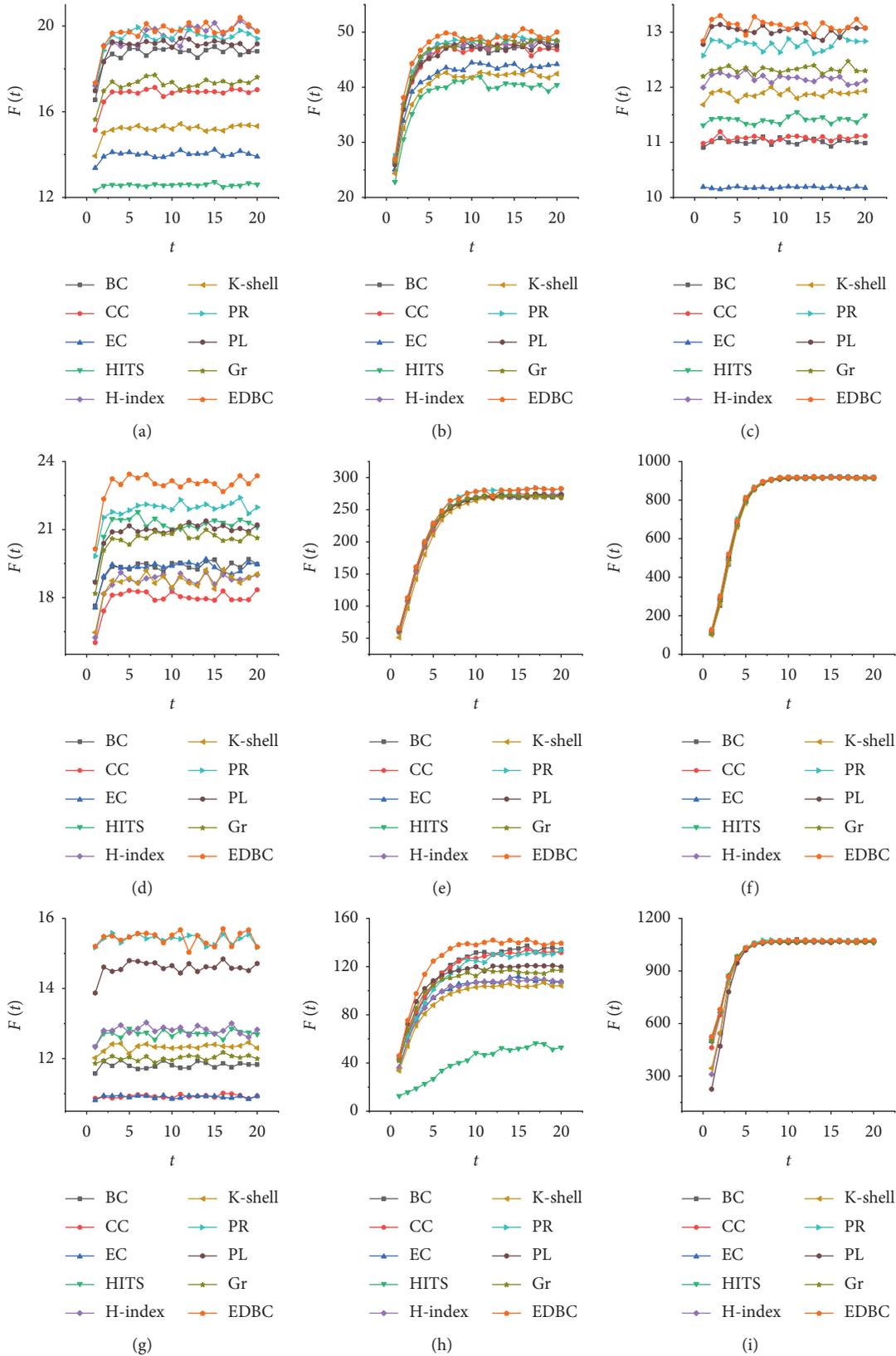


FIGURE 5: Influence of propagation for top-10 ranking effects using ten various algorithms, where $F(t)$ represents the number of infected and recovered nodes at time (t) in a variable range between 0 and 20. (a) Physicians. (b) e-mail. (c) Subelij-euroroad. (d) Air-traffic control. (e) Petster-friendships. (f) Web-Spam. (g) US-Powergrid. (h) PGP algorithm. (i) CAIDA-project, etc.

5. Conclusion and Future Recommendations

This paper investigates the problem of influential or key node ranking from a performance evaluation's viewpoint. In view of related studies, several methods have been offered to explore how to detect the key or influential node-based centrality measures. However, these centrality measures' algorithms have their own advantages and disadvantages. In this study, we proposed an effective-based centrality method for detecting the influential nodes in complex networks. Our proposed algorithm considered K-shell, degree nodes, effective distance, and several levels of neighbor's influence or neighborhood potential. Consequently, this algorithm can be employed in any network, i.e., directed or undirected. In order to evaluate the performance and effectiveness of the proposed algorithm, we applied several types of real-world networks and used two standard evaluation criteria, SIR and Kendall's correlation coefficient τ methods, to analyze the spreading influence of ranking nodes. The experimental setup demonstrated that the proposed algorithm regarding accuracy and effectiveness is reasonable and significant as compared to the classical sorting algorithms and recently proposed several relevant algorithms. However, there persist challenging issues that need to be addressed to the quality extent of current work. For instance, adding more parameters to adjust the intensity between the nodes to yield better performance is a challenge. Onward, we plan to improve the proposed algorithm in consideration of weighted formal-concept analysis.

Data Availability

All relevant data are publicly available at <http://konect.uni-koblenz.de/networks/>, <http://snap.stanford.edu/data/as-caida.html> and <http://networkrepository.com/web-spam.php>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant no. 2018YFB1003602.

References

- [1] J. Zhao, Y. Wang, and Y. Deng, "Identifying influential nodes in complex networks from global perspective," *Chaos, Solitons & Fractals*, vol. 133, Article ID 109637, 2020.
- [2] N. Zhao, J. Bao, and N. Chen, "Ranking influential nodes in complex networks with information entropy method," *Complexity*, vol. 2020, Article ID 5903798, 15 pages, 2020.
- [3] J. Dai, B. Wang, J. Sheng et al., "Identifying influential nodes in complex networks based on local neighbor contribution," *IEEE Access*, vol. 7, pp. 131719–131731, 2019.
- [4] X. Chen, M. Tan, J. Zhao, T. Yang, D. Wu, and R. Zhao, "Identifying influential nodes in complex networks based on a spreading influence related centrality," *Physica A: Statistical Mechanics and Its Applications*, vol. 536, Article ID 122481, 2019.
- [5] S. Wang, Y. Du, and Y. Deng, "A new measure of identifying influential nodes: efficiency centrality," *Communications in Nonlinear Science and Numerical Simulation*, vol. 47, pp. 151–163, 2017.
- [6] F. Schweitzer, G. Fagiolo, D. Sornette, F. Vega-Redondo, A. Vespignani, and D. R. White, "Economic networks: the new challenges," *Science*, vol. 325, no. 5939, p. 422, 2009.
- [7] R. C. T. M. Garas and H. S. Argyrakis, "Worldwide spreading of economic crisis," *New Journal of Physics*, vol. 12, no. 11, 2010.
- [8] U. Alon, "Biological networks: the tinkerer as an engineer," *Science*, vol. 301, no. 5641, p. 1866, 2003.
- [9] R. Milo and S. Shen-Orr, "Network motifs: simple building blocks of complex networks," *Science*, vol. 298, no. 5594, p. 824, 2002.
- [10] H. Yu, X. Cao, Z. Liu, and Y. Li, "Identifying key nodes based on improved structural holes in complex networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 486, pp. 318–327, 2017.
- [11] J. Liu, Q. Xiong, W. Shi, X. Shi, and K. Wang, "Evaluating the importance of nodes in complex networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 452, pp. 209–219, 2016.
- [12] J. Liu, F. Lian, and M. Mallick, "Distributed compressed sensing based joint detection and tracking for multistatic radar system," *Information Sciences*, vol. 369, pp. 100–118, 2016.
- [13] A. Sheikahmadi, M. A. Nematbakhsh, and A. Shokrollahi, "Improving detection of influential nodes in complex networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 436, pp. 833–845, 2015.
- [14] D. Y. Kenett, M. Perc, and S. Boccaletti, "Networks of networks - an introduction," *Chaos, Solitons & Fractals*, vol. 80, pp. 1–6, 2015.
- [15] T. Bian and Y. Deng, "Identifying influential nodes in complex networks: a node information dimension approach," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 4, Article ID 043109, 2018.
- [16] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, no. 4, pp. 581–603, 1966.
- [17] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972.
- [18] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.
- [19] E. Estrada and N. Hatano, "Resistance distance, information centrality, node vulnerability and vibrations in complex networks," in *Network Science: Complexity in Nature and Technology*, p. 13, Springer, London, UK, 2010.
- [20] M. E. J. Newman, "Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality," *Physical Review E*, vol. 64, Article ID 016132, 2001.
- [21] E. Estrada and J. A. Rodríguez-Velázquez, "Subgraph centrality in complex networks," *Physical Review E*, vol. 71, Article ID 056103, 2005.
- [22] S. Brin and L. Page, "Reprint of: the anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 56, no. 18, pp. 3825–3833, 2012.
- [23] D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, and T. Zhou, "Vital nodes identification in complex networks," *Physics Reports*, vol. 650, pp. 1–63, 2016.

- [24] M. Kitsak, L. K. Gallos, S. Havlin et al., *Identification of Influential Spreaders In Complex Networks*, 2010.
- [25] J. M. Kleinberg, “Authoritative sources in a hyperlinked environment,” *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.
- [26] L. Lv, K. Zhang, T. Zhang, D. Bardou, J. Zhang, and Y. Cai, “Pagerank centrality for temporal networks,” *Physics Letters A*, vol. 383, no. 12, pp. 1215–1222, 2019.
- [27] M. Kitsak, L. K. Gallos, S. Havlin et al., “Identification of influential spreaders in complex networks,” *Nature Physics*, vol. 6, no. 11, p. 888, 2010.
- [28] B. L. H. S. Shao and S. H. E. Buldyrev S, “Structure of shells in complex networks,” *Physical Review E*, vol. 80, no. 3, 2009.
- [29] B. A. Alvarez-Hamelin and V. A. Dall’Asta, “K-core decomposition of internet graphs: hierarchies, self-similarity and measurement biases,” *NHM*, vol. 3, no. 2, p. 371, 2008.
- [30] J. E. Cohen, “Infectious diseases of humans: dynamics and control,” *JAMA: The Journal of the American Medical Association*, vol. 268, no. 23, p. 3381, 1992.
- [31] M. G. Kendall, “A new measure of rank correlation,” *Biometrika*, vol. 30, no. 1-2, pp. 81–93, 1938.
- [32] O. Fercoq, M. Akian, M. Bouhtou, and S. Gaubert, “Ergodic control and polyhedral approaches to pagerank optimization,” *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 134–148, 2013.
- [33] Z. Jovanović, E. Milovanović, and I. Milovanović, “Some remarks on laplacian eigenvalues of connected graphs,” *Linear Algebra and Its Applications*, vol. 503, pp. 48–55, 2016.
- [34] C. Canali and R. Lancellotti, “A quantitative methodology based on component analysis to identify key users in social networks,” *International Journal of Social Network Mining*, vol. 1, no. 1, pp. 27–50, 2012.
- [35] R. Elhesha, T. Kahveci, and B. Baiser, “Motif centrality in food web networks,” *Journal of Complex Networks*, vol. 5, no. 4, pp. 641–664, 2017.
- [36] Z. Yu, J. Shao, Q. Yang, and Z. Sun, “Profliteader: identifying leaders in networks with profit capacity,” *World Wide Web*, vol. 22, no. 2, pp. 533–553, 2019.
- [37] Z.-K. Bao, J.-G. Liu, and H.-F. Zhang, “Identifying multiple influential spreaders by a heuristic clustering algorithm,” *Physics Letters A*, vol. 381, no. 11, pp. 976–983, 2017.
- [38] H. Mo, C. Gao, and Y. Deng, “Evidential method to identify influential nodes in complex networks,” *Journal of Systems Engineering and Electronics*, vol. 26, no. 2, p. 381, 2015.
- [39] T. Bian and Y. Deng, “A new evidential methodology of identifying influential nodes in complex networks,” *Chaos, Solitons & Fractals*, vol. 103, pp. 101–110, 2017.
- [40] T. Bian, J. Hu, and Y. Deng, “Identifying influential nodes in complex networks based on ahp,” *Physica A: Statistical Mechanics and Its Applications*, vol. 479, pp. 422–436, 2017.
- [41] A. Zeng and C.-J. Zhang, “Ranking spreaders by decomposing complex networks,” *Physics Letters A*, vol. 377, no. 14, pp. 1031–1035, 2013.
- [42] J.-H. Lin, Q. Guo, W.-Z. Dong, L.-Y. Tang, and J.-G. Liu, “Identifying the node spreading influence with largest k-core values,” *Physics Letters A*, vol. 378, no. 45, pp. 3279–3284, 2014.
- [43] J. Bae and S. Kim, “Identifying and ranking influential spreaders in complex networks by neighborhood coreness,” *Physica A: Statistical Mechanics and Its Applications*, vol. 395, pp. 549–559, 2014.
- [44] L.-l. Ma, C. Ma, H.-F. Zhang, and B.-H. Wang, “Identifying influential spreaders in complex networks based on gravity formula,” *Physica A: Statistical Mechanics and Its Applications*, vol. 451, pp. 205–212, 2016.
- [45] Y. Liu, B. Wei, Y. Du, F. Xiao, and Y. Deng, “Identifying influential spreaders by weight degree centrality in complex networks,” *Chaos, Solitons & Fractals*, vol. 86, pp. 1–7, 2016.
- [46] Y. Liu, M. Tang, T. Zhou, and Y. Do, “Identify influential spreaders in complex networks, the role of neighborhood,” *Physica A: Statistical Mechanics and Its Applications*, vol. 452, pp. 289–298, 2016.
- [47] J. Wang, X. Hou, K. Li, and Y. Ding, “A novel weight neighborhood centrality algorithm for identifying influential spreaders in complex networks,” *Physica A: Statistical Mechanics and Its Applications*, vol. 475, pp. 88–105, 2017.
- [48] B. Wei, J. Liu, D. Wei, C. Gao, and Y. Deng, “Weighted k-shell decomposition for complex networks based on potential edge weights,” *Physica A: Statistical Mechanics and Its Applications*, vol. 420, pp. 277–283, 2015.
- [49] P. M. Gleiser and L. Danon, “Community structure in jazz,” *Advances in Complex Systems*, vol. 06, no. 04, pp. 565–573, 2003.
- [50] C. Gao, X. Lan, X. Zhang, and Y. Deng, “A bio-inspired methodology of identifying influential nodes in complex networks,” *PLoS One*, vol. 8, no. 6, 2013.
- [51] D. Gómez, J. R. Figueira, and A. Eusébio, “Modeling centrality measures in social network analysis using bi-criteria network flow optimization problems,” *European Journal of Operational Research*, vol. 226, no. 2, pp. 354–365, 2013.
- [52] W. Zang, P. Zhang, C. Zhou, and L. Guo, “Locating multiple sources in social networks under the sir model: a divide-and-conquer approach,” *Journal of Computational Science*, vol. 10, pp. 278–287, 2015.
- [53] L. Fei, Q. Zhang, and Y. Deng, “Identifying influential nodes in complex networks based on the inverse-square law,” *Physica A: Statistical Mechanics and Its Applications*, vol. 512, pp. 1044–1059, 2018.
- [54] D. Brockmann and D. Helbing, “The hidden geometry of complex, network-driven contagion phenomena,” *Science*, vol. 342, no. 6164, pp. 1337–1342, 2013.
- [55] V. Batagelj and M. Zaveršnik, “Fast algorithms for determining (generalized) core groups in social networks,” *Advances in Data Analysis and Classification*, vol. 5, no. 2, pp. 129–145, 2011.
- [56] J. Coleman, E. Katz, and H. Menzel, “The diffusion of an innovation among physicians,” *Sociometry*, vol. 20, no. 4, pp. 253–270, 1957.
- [57] R. Guimer, L. Danon, A. Daz-Guilera, F. Giralt, and A. Arenas, “Self-similar community structure in a network of human interactions,” *Physical Review E*, vol. 68, no. 6, Article ID 065103, 2003.
- [58] L. Šubelj and M. Bajec, “Robust network community detection using balanced propagation,” *European Physical Journal B*, vol. 81, no. 3, pp. 353–362, 2011.
- [59] Aviation Administration, “Air traffic control system command center,” 2017, <http://www.fly.faa.gov/>.
- [60] Hamsterster friendships network dataset-KONECT (Apr. 2017), <http://konect.uni-koblenz.de/networks/petster-friendships-hamster>.
- [61] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, no. 1, pp. 440–442, 1998.
- [62] C. Castillo, K. Chellapilla, and L. Denoyer, “Web spam challenge,” in *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb, Beijing, China, 2008.

- [63] M. Bogu, R. Pastor-Satorras, A. Daz-Guilera, and A. Arenas, "Models of social networks based on social distance attachment," *Physical Review E*, vol. 70, no. 5, Article ID 056122, 2004.
- [64] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: densification and shrinking diameters," *ACM Trans. Knowledge Discovery from Data*, vol. 1, no. 1, pp. 1–40, 2007.
- [65] L. J. Allen, "Some discrete-time si, sir, and sis epidemic models," *Mathematical Biosciences*, vol. 124, no. 1, pp. 83–105, 1994.
- [66] Y. Gang, Z. Tao, W. Jie, F. Zhong-Qian, and W. Bing-Hong, "Epidemic spread in weighted scale-free networks," *Chinese Physics Letters*, vol. 22, no. 2, p. 510, 2005.
- [67] M. G. Kendall, "The treatment of ties in ranking problems," *Biometrika*, vol. 33, no. 3, pp. 239–251, 1945.
- [68] W. R. Knight, "A computer method for calculating kendall's tau with ungrouped data," *Journal of the American Statistical Association*, vol. 61, no. 314, pp. 436–439, 1966.
- [69] Y. Yang, X. Wang, Y. Chen, and M. Hu, "Identifying key nodes in complex networks based on global structure," *IEEE Access*, vol. 8, pp. 32904–32913, 2020.
- [70] M. Kitsak, L. K. Gallos, S. Havlin et al., "Identification of influential spreaders in complex networks," *Nature Physics*, vol. 6, no. 11, pp. 888–893, 2010.