



Article A Novel Method to Identify Key Nodes in Complex Networks Based on Degree and Neighborhood Information

Na Zhao ^{1,2,3}, Shuangping Yang ¹, Hao Wang ¹, Xinyuan Zhou ¹, Ting Luo ¹ and Jian Wang ^{4,*}

- ¹ Key Laboratory in Software Engineering of Yunnan Province, School of Software, Yunnan University, Kunming 650091, China; zhaona@ynu.edu.cn (N.Z.); yangshuangping@mail.ynu.edu.cn (S.Y.); wh873236094@163.com (H.W.); starcoming@mail.ynu.edu.cn (X.Z.); 18979213005@163.com (T.L.)
- ² Big Data Research Center, University of Electronic Science and Technology of China, Chengdu 610056, China
- ³ The Key Laboratory for Crop Production and Smart Agriculture of Yunnan Province, Kunming 650201, China
- Faculty of Information Engineering and Automation, Kunming University of Science and Technology,
- Kunming 650504, China
- * Correspondence: jianwang@kust.edu.cn

Abstract: One key challenge within the domain of network science is accurately finding important nodes within a network. In recent years, researchers have proposed various node centrality indicators from different perspectives. However, many existing methods have their limitations. For instance, certain approaches lack a balance between time efficiency and accuracy, while the majority of research neglects the significance of local clustering coefficients, a crucial node property. Thus, this paper introduces a centrality metric called DNC (degree and neighborhood information centrality) that considers both node degree and local clustering coefficients. The combination of these two aspects provides DNC with the ability to create a more comprehensive measure of nodes' local centrality. In addition, in order to obtain better performance in different networks, this paper sets a tunable parameter α to control the effect of neighbor information on the importance of nodes. Subsequently, the paper proceeds with a sequence of experiments, including connectivity tests, to validate the efficacy of DNC. The results of the experiments demonstrate that DNC captures more information and outperforms the other eight centrality metrics.

Keywords: key nodes; complex network; robustness; local clustering coefficient

1. Introduction

As science and technology have advanced, it has become clear that numerous natural phenomena can be effectively described using network models [1]. Various systems in the physical world, such as interpersonal relationships on social media [2], protein–protein interactions within biological organisms [3], and even road network [4], can be abstracted as complex networks.

Identifying key nodes in a network holds significant practical relevance [5]. For instance, in social networks, certain individuals may serve as crucial mediators for information propagation. In transportation networks, specific intersection points can significantly influence the entire network's fluidity. In the context of disease spread research, identifying key propagators can aid in controlling the diffusion of infectious disease [6]. In urban planning, uncovering key nodes within transportation networks can enhance a city's resilience and sustainability [7]. Furthermore, in the realms of social media and Internet advertising, determining key nodes in advertising dissemination can assist in optimizing advertising strategies [8].

Different types of networks and physical systems necessitate different approaches to determining node importance. A number of traditional algorithms have been proposed by researchers. These classical centrality algorithms have found widespread applications. Over the past few decades, many efficient methods have been developed to assess the



Citation: Zhao, N.; Yang, S.; Wang, H.; Zhou, X.; Luo, T.; Wang, J. A Novel Method to Identify Key Nodes in Complex Networks Based on Degree and Neighborhood Information. *Appl. Sci.* 2024, *14*, 521. https://doi.org/ 10.3390/app14020521

Academic Editors: Giacomo Fiumara, Xiaoyang Liu, Pasquale De Meo and Annamaria Ficara

Received: 27 November 2023 Revised: 1 January 2024 Accepted: 5 January 2024 Published: 7 January 2024



Copyright: © 2024 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/). nodes' importance by leveraging both local and global information, along with other relevant attributes. For example, EnRenew [9] utilizes local entropy to pinpoint key nodes within networks; Spon [10] assesses node importance using information from neighboring nodes; VoteRank [11] employs a voting mechanism to recognize significant disseminators in networks; and FINDER [12] utilizes reinforcement learning to identify crucial nodes within networks.

Current investigations into extracting critical information from complex networks have frequently presented researchers with the challenge of reconciling method efficiency with precision [10]. Therefore, based on degree and neighbor node information, this paper introduces an innovative approach to identify crucial nodes within networks, referred to as the degree and neighborhood information centrality (DNC). DNC contends that a node's importance within the network relies on its personal information and the information provided by adjacent nodes. The node's individual information is its degree, while the information from neighboring nodes is determined by the combined local clustering coefficients of its neighbors. The following are the primary advantages of our proposed DNC:

- (1) Low time complexity: Since it only utilizes a node's degree and first-order information, DNC only necessitates the computation of first-order neighbor information, leading to a temporal complexity of O(|E|), allowing DNC to be effectively applied to large-scale networks.
- (2) High accuracy: DNC surpasses existing baseline methods in terms of accuracy.
- (3) Parameter setting: DNC can be parameterized to adapt to different networks for optimal performance.

The sections of this paper that follow are ordered in a logical order. Section 2 includes a synopsis of relevant work, encompassing classical node importance algorithms and recent advancements. Following this, in Section 3, we introduce our baseline methods and proceed with an in-depth presentation of our proposed algorithm, including a demonstration using a small dataset. Section 4 outlines the comprehensive procedure and specifics of our experiments. Finally, we conclude with an overview of the topics covered in this paper.

2. Related Works

In the past few years, critical nodes in complex networks have received special attention from researchers. We present a quick review of current algorithms in this section.

Researchers have presented a variety of classical centrality measurements based on diverse concepts and ideas. Degree centrality [13] quantifies the number of neighbors a node has. Although degree centrality calculations are straightforward, they frequently overlook the overall structure of complex networks, resulting in diminished accuracy. Eigenvector centrality [14] computes a node's score by considering the centrality of its neighbors. Closeness centrality [15] evaluates the average distance from the present node to all other nodes along the shortest paths. Betweenness centrality [13] measures a node's significance by allying shortest paths that traverse through it, reflecting its significance as a bridge node. The VoteRank algorithm [11] employs a voting mechanism to assign scores to nodes.

The fundamental concept behind the K-Shell algorithm involves identifying core nodes within a network by iteratively eliminating nodes with the lowest degrees [16]. Yet, the K-Shell algorithm grapples with several limitations; it predominantly focuses on node degrees, disregards other pertinent network attributes, and is unable to differentiate the score of nodes within the same layer. Based on the K-Shell concept, Zareie et al. [17] suggested a customized hierarchical method, augmenting it with more topological information. Wang et al. [18] proposed an enhanced K-Shell algorithm that identifies important nodes from higher-level K-Shells to lower-level K-Shells using node information entropy. A mixed-degree decomposition approach was developed by Zeng et al. [19]. This method integrates the concept of depletion degree to assess the impact of eliminated nodes on a specific node, as well as remaining degree to assess the impact of remaining nodes on that specific node.

Furthermore, methods for identifying important nodes based on improved gravity models have also gained significant attention [20–24]. Inspired by the gravity model, node degrees are considered as masses, and the distance is represented by the shortest path between two nodes. The importance score of nodes is then calculated using the formula for universal gravitation [20]. However, in practical usage, the gravity model is associated with high time complexity, and using node degree as mass often lacks accuracy due to the limited information node degrees represent [21]. Liu et al. introduced weighted gravity centrality [22], which combines eigenvector centrality with the gravity model, integrating global and local information. Yang et al. introduced KSGC, which added an attraction coefficient to the K-Shell-based generalized gravity model [21]. Li et al. proposed MCGM [23], considering more node information and combining node degree, K-Shell, and EC as node masses. Recently, Xu et al. proposed a novel communication-based adaptive gravity model, CAGM, an innovative communication-based adaptive gravity model that assesses each node's score by using the likelihood and intensity of influence from nearby nodes within its impact radius [24].

Researchers have also been devoted to using deep learning to identify important nodes in complex networks [12,25,26]. Yu et al. proposed employing graph convolutional networks to locate key nodes [25]. Using graph convolutional networks, Ou et al. [26] developed a method for finding critical nodes while accounting for different levels of network structural features. Fan et al. introduced FINDER, a strategy based on reinforcement learning for finding key nodes [12].

3. Causality in Complex Networks

Measuring causality in complex networks is crucial for understanding system behavior, optimizing network design, and addressing practical challenges. It helps reveal mutual influences between nodes, providing profound insights into network dynamics, robustness, and other properties, fostering the advancement of scientific research and practical applications [27].

3.1. Limitations of Statistical Approaches in Unveiling and Discovering Causality in Complex Networks

Understanding a system's basic structure, dynamics, and relationships requires measuring causality in a network. When dealing with causal interactions in complicated networks, traditional statistical methods have several drawbacks. These methods often rely on linear regression and correlation tests, both of which have modeling limitations. Traditional causal inference methods rely primarily on probability distributions and may produce incorrect conclusions. Traditional statistical methods sometimes encounter difficulties with hierarchical organization and inductive reasoning [27]. Some classic methods neglect non-statistical aspects, rely too heavily on randomness, and struggle to deal with complex systems. These techniques frequently concentrate exclusively on correlations between variables, which may not imply causation. Traditional techniques have major hurdles when dealing with non-ordered data and large-scale high-dimensional data [28].

3.2. The Difference between DNC and Approaches That Try to Infer Causal Structures

DNC is mostly based on node topological characteristics, with an emphasis on measuring a node's connection and clustering in the network. Statistical models, probabilistic graphical models, or causal inference methodologies are commonly used to infer causal structure. These methods frequently take into account causal linkages between variables rather than just topological structure. DNC's purpose is to measure nodes, with a focus on node centrality and social aggregation. Causal inference approaches seek to comprehend the causal links between variables, that is, how changes in one variable cause changes in another. DNC focuses on network topological structure analysis, whereas causal structure inference methods are more concerned with understanding causal relationships between variables. These two approaches differ in problem orientation, data requirements, and application domains.

3.3. Classical Entropy Measures in Networks

Entropy measurement in complex networks reveals the complexities of network topology and information dispersion, providing insights into node relevance and network dynamics. It provides critical data for network design and optimization.

Traditional entropy measurements are heavily influenced by the feature descriptions used, such as adjacency matrices or degree sequences, limiting them to basic counting functions and hindering independent assessments of object randomness. Entropy is extremely sensitive to the observer's point of view, which can result in differing values under different descriptions, leading to misleading results. Entropy is largely dependent on probability distributions, and different distribution choices might result in varying entropy values, creating uncertainty when assessing network complexity. While entropy is a computational metric, it may not accurately reflect the underlying complexity of highly complex network architectures. In comparison, algorithmic complexity measurement is more robust, especially when describing non-random, recursively created networks. Entropy confronts difficulties when it comes to capturing network uniqueness [29].

4. Proposed Method

In this section, we will begin by reviewing the centrality metrics related to the experiments performed herein. Subsequently, we will present a full explanation of DNC.

4.1. Baseline Methods

4.1.1. Collective Influence (CI)

The fundamental concept behind collective influence (CI) is to evaluate the score of a node by evaluating how its removal impacts the giant connected components of the network [30]. CI can be represented as:

$$\operatorname{CI}_{\ell}(i) = (k_i - 1) \sum_{j \in \partial \operatorname{Ball}(i,\ell)} (k_j - 1)$$
(1)

where k_i indicates node *i*'s degree, Ball (i, ℓ) signifies all nodes within a ball centered on node *i* with a radius of ℓ , and ℓ is a preset value.

4.1.2. K-Shell (KS)

Kitsak et al. [16] introduced the K-Shell, which estimates the relevance of nodes based on their network placements. The following are the steps involved in K-Shell decomposition: initially, delete all network nodes and edges with degrees less than or equal to 1. Moreover, any subsequently generated nodes with a degree of 1, along with their respective edges, are also removed. This cycle continues until no nodes with degrees equal to or less than one remain. The nodes removed throughout this process constitute the 1-shell layer. The method then iterates to remove those nodes with degrees less than or equal to 2, resulting in the formation of the 2-shell layer. This procedure is performed until all nodes are removed.

4.1.3. Closeness Centrality (CC)

The core idea of closeness centrality (CC) is that the influence of a node is related to the shortest path length from it to other nodes, with smaller average shortest path lengths indicating greater influence. *CC* can be represented as:

$$CC(i) = \frac{N-1}{\sum_{j \neq i} d_{ij}}$$
(2)

where d_{ij} is the shortest path distance between node *i* to node *j*.

4.1.4. K-Shell Gravity Centrality (KSGC)

Yang et al. introduced the KSGC algorithm, an enhanced version of the gravity model algorithm rooted in K-Shell. This approach introduces a gravitational coefficient to depict the interaction force between nodes, encompassing considerations of node position and local and global information [21]. KSGC can be represented as:

$$F(i,j) = c_{ij} \frac{k_i * k_j}{d_{ij}^2}$$
(3)

The variable $c_{ij} = e^{\frac{ks(i)-ks(j)}{ks_{max}-ks_{min}}}$, where ks(i) and ks(j) denote the K-Shell values of the node, while ks_{max} and ks_{min} represent the maximum and minimum K-Shell values in the network.

4.1.5. Local Version of GM (LGM)

Li et al. proposed LGM, which leverages neighborhood information and path details and incorporates a localized gravity model featuring a truncation radius. This approach aims to identify significant nodes within complex networks [31]. LGM can be represented as:

$$LGM(i) = \sum_{d_{ij} \leqslant R, j \neq i} \frac{k_i * k_j}{d_{ij}^2}$$
(4)

where *R* represents the truncation radius.

4.1.6. Laplacian Gravity Centrality (LGC)

Zhang et al. introduced LGC, which optimizes the initial gravity centrality by incorporating the Laplacian centrality as mass, taking into consideration the degrees of adjacent nodes [32]. LC can be represented as:

$$LC(i) = k_i^2 + k_i + 2\sum_{j \in \Gamma_i} k_j$$
(5)

where LC(i) represents the Laplacian centrality of node *i* and Γ_i is the set of nodes that are neighboring node *i*.

Then, by adding Laplacian gravity within the cutoff radius of topological distance pairs, LGC can be represented as:

$$LGC(i) = \sum_{j \neq i, d_{ij} \le \langle d \rangle/2} \frac{LC(i)LC(j)}{d_{ij}^2}$$
(6)

where $\langle d \rangle$ is the average topological distance between network nodes.

4.1.7. Social Capital (SC)

Zhou et al. introduced social capital (SC), suggesting that nodes with higher degrees and the cumulative degrees of their neighbors hold crucial positions within the network [33]. It is defined as:

$$s_i = k_i + \sum_{j \in \Gamma_i} k_j \tag{7}$$

4.1.8. Improved Gravitational Centrality (IGC)

Wang et al. proposed improved gravitational centrality, which is an enhanced gravitational centrality method that combines K-Shell values and degrees. This approach uses the

node's K-Shell value as mass and considers the degrees as mass for adjacent nodes [34]. The algorithm is defined as follows:

$$G_{+}(i) = \sum_{j \in \Gamma_i} \sum_{p \in \Psi_j} \frac{ks(j) * k_p}{d_{jp}^2}$$
(8)

where Ψ_i adjacent nodes whose shortest path length is less than the specified length.

4.2. DNC Method

4.2.1. Local Clustering Coefficient

The local clustering coefficient of a node is an important metric that measures the level of closeness between nodes in a network, delineating the density of connections among a node's neighboring nodes.

For unweighted graphs, a node's local clustering coefficient is defined as the ratio of actual existing edges between neighboring nodes to the potential number of edges between them [35]. It is defined as follows:

$$C_i = \frac{2 \cdot E_i}{k_i \cdot (k_i - 1)} \tag{9}$$

where E_i denotes the number of edges connecting the node to its neighbor.

4.2.2. DNC Method

The DNC method consists of two main steps. First, it calculates the sum of the local clustering coefficients of the first-order neighbors of a node, denoted as slc_i . Then, the node's DNC value is obtained by adding its degree to the sum of the first-order neighbor local clustering coefficients. The slc_i is defined as follows:

$$slc_i = \sum_{j \in \Gamma_i} c_j$$
 (10)

where c_i represents the local clustering coefficient of node *i*.

Subsequently, the *DNC* value of a node is computed as follows:

$$DNC_i = k_i + \alpha * slc_i \ (\alpha > 0) \tag{11}$$

where α denotes an adjustable parameter to control the effect of *slc* on the calculation results. In this paper, $\alpha = 1$ is used for the calculation.

DNC considers both the node's own information and the node's neighbors' information, that is, the sum of the local clustering coefficients of the first-order neighbors. It measures the density of connections among its neighbors. By considering the sum of first-order neighbor local clustering coefficients, this method can comprehensively capture the local network structure between the node and its most immediate associates, providing insights beyond the node's individual information. By adding the node's self-information, it can more accurately estimate the node's importance within its neighborhood. This prevents highly connected nodes (those with numerous neighbors) from being overly emphasized in importance assessment while underrating nodes with low degrees (having fewer neighbors). A person's influence in social networks is frequently connected to their position in the social network and their ties with first-order neighbors. The sum of the first-order neighbor local clustering coefficients can better reflect an individual's influence in the social network.

The pseudocode for DNC is provided in Algorithm 1:

Algorithm 1: The proposed DNC method's framework
Input: Graph $G = (V, E), V(G) = N.$
Output: Node importance ranking list: $\{DNC(u) \mid u \in V\}$
1: for $u \in V$ do
2: Calculate C(<i>u</i>) using Equation (9)
3: end for
4: for $u \in V$ do
5: $FN \leftarrow u.$ neighbor
6: for fn in FN do
7: $slc(u) \leftarrow sum(C(fn))$
8: end for
9: $DNC(u) \leftarrow slc(u) + k(u)$
10: end for
11: Sort the list $\{DNC(u) u \in V\}$ in descending order
12: return the ranking list of nodes

4.2.3. Example Analysis

We use the karate network [36] to illustrate the execution process of the DNC algorithm. The karate network is made up of social contacts between Zachary Karate Club members, as shown in Figure 1, and it is a network with 34 nodes and 78 edges. We initiate the demonstration of the DNC by selecting node 8 as an example to showcase its execution process. Subsequently, we compute the top ten nodes identified by DNC and perform a comparative analysis with the baseline methods.



Figure 1. Karate network. The numbers in the figure indicate the node numbers.

First, we calculate the degree of node k_8 , which is 5. Next, we identify the neighboring nodes of node 8, which are its closest neighbors 0, 2, 30, 32, and 33.

Next, we calculate the local clustering coefficient (LCC) for each neighboring node. Subsequently, we compute the *slc* of node 8. The LCC for the neighboring nodes of node 8 are as shown in Table 1.

Table 1. The LCC of the first-order neighbors of node 8 in the Karate network.

Node	0	2	30	32	33
LCC	0.15	0.24	0.50	0.20	0.11

Finally, we obtain the DNC value of this node by adding its degree and the *slc*.

$$DNC(8) = k_8 + \sum_{j \in N_8} c_j = 5 + (0.15 + 0.24 + 0.50 + 0.20 + 0.11) = 6.2$$
(12)

We present the top ten ranked nodes using various methods, as shown in Table 2. The first row represents the method names.

DNC	CI	KS	CC	KSGC	LGM	LGC	SC	IGC
33	0	0	0	8	33	33	0	0
0	33	1	2	13	0	0	33	33
32	32	2	33	30	32	32	2	2
1	2	3	31	7	2	2	32	32
2	1	7	8	3	1	1	8	1
3	31	8	13	19	31	8	13	3
31	3	13	32	31	8	13	1	8
13	8	30	19	19	13	31	31	13
23	13	32	1	23	3	23	3	31
5	30	33	3	27	23	3	30	30
33	0	0	0	8	33	33	0	0
0	33	1	2	13	0	0	33	33
32	32	2	33	30	32	32	2	2
1	2	3	31	7	2	2	32	32

Table 2. Top ten nodes with DNC compared to baseline methods.

From the table above, we can observe that the top three nodes ranked by DNC are identical to those of LGM and LGC. There are slight variations in the ordering of the top five nodes between DNC, CI, LGM, and LGC. In the following sections, we will use Kendall's Tau (τ) to quantify the associations between the sequences generated by different methods.

4.2.4. Time Complexity

When utilizing the DNC to compute the score of nodes in a network, it is necessary to compute the degree of nodes as well as the sum of their first-order neighbors' local clustering coefficients. First, calculate the local clustering coefficient for all nodes in the network. The computation of local clustering coefficients involves traversing the neighbors of nodes and calculating their connections, making the time complexity for calculating the local clustering coefficients for all nodes in a network O(|E|). To calculate the importance score for each node in the network, as it involves traversing the first-order neighbors of nodes, the time complexity for this part is $O(\bar{k})$, where \bar{k} is the average degree of nodes. In a network, |E| is significantly larger than \bar{k} , which results in the overall time complexity of DNC being O(|E|).

5. Experiment Conclusions

5.1. Datasets

We tested the performance of DNC using 12 empirical unweighted networks, including the following:

- (1) Dolphins: A record of interactions involving 62 dolphins [37].
- (2) Polbooks: Political books associated with the 2004 presidential elections are available for purchase on Amazon [38].
- (3) Adjnoun: A network constructed by recording and analyzing relationships between adjectives and nouns in text [39].
- (4) Jazz: Jazz musicians collaborate, perform, record, and engage in compositional activities, forming a network that documents these interactions [40].
- (5) C_elegans: Documenting and analyzing the neural system connections and interactions of Caenorhabditis elegans, a type of roundworm [35].

- (6) USAir97: The network of the United States' air transportation system [38].
- (7) Vote: A social network dataset representing voting relationships among Wikipedia users [41].
- (8) Email: A communication network representing internal email exchanges at a university [42].
- (9) Yeast: Describing the interactions of proteins within yeast cells [43].
- (10) Hamsterster: A website network that connects user friendships and family relationships [38].
- (11) Kohonen: A network based on the neural network model of self-organizing maps, initially proposed for the citation network analysis in Pajek [44].
- (12) Dmela: A network studying protein–protein interaction relationships and their impact on the biological processes and functions in the fruit fly Drosophila melanogaster [38]. The basic topological features of the network number are shown in Table 3.

Table 3. The typical metric values for these 12 networks. The structural parameters and topological properties of all of these networks, including the number of nodes N and edges E, average degree $\langle K \rangle$, maximum degree K_{max} , clustering coefficient C, and assortativity coefficient r.

Network	N	Е	<k></k>	K _{max}	С	r
Dolphins	62	159	5	12	0.2590	-0.0436
Polbooks	105	441	8	25	0.4875	-0.1279
Adjnoun	112	425	7	49	0.1728	-0.1293
Jazz	198	2742	27	100	0.5203	0.0202
C_elegans	297	2148	15	134	0.3115	-0.1520
USAir97	332	2126	12	139	0.6252	-0.2079
Vote	889	2914	7	102	0.1528	-0.0288
Email	1133	5451	9	71	0.2200	0.0782
Yeast	2375	11,693	9	118	0.3100	0.4539
Hamsterster	2426	16,630	14	273	0.5375	0.0474
Kohonen	3772	12,718	5	740	0.2100	-0.1204
Dmela	7393	25,569	6	190	0.0118	-0.0465

5.2. Metrics

5.2.1. Robustness Metric

Connectivity testing stands as one of the classical validation methods for assessing the importance of nodes within a network [10]. The importance sequence of nodes is established through a designated algorithm, after which these nodes are systematically removed one by one. The algorithm's accuracy is measured by the extent of network collapse after each node deletion. We use the robustness index *R* to describe the variation in network collapse upon node removal [45], denoted as:

$$R = \frac{1}{N} \sum_{i=1}^{N} \sigma(i/N) \tag{13}$$

N denotes the number of network nodes. Utilizing a normalization factor of 1/N allows for the comparison of robustness across networks of different sizes. *R* values range from 1/N to 0.5. Furthermore, $\sigma(i/N)$ signifies the ratio of nodes within the largest remaining connected component post node elimination in comparison to the overall number of nodes in the graph. We measured the accuracy of DNC and the baseline methods by plotting the change in $\sigma(i/N)$ after node removal. The region enclosed by the X and Y axes and the curve is used to determine the robustness index *R*. If the area enclosed by the axes and the curve is minimized, and $\sigma(i/N)$ decreases the fastest, this suggests that the network crumbles the fastest and the sequence of important nodes ranked by this method is the most accurate.

5.2.2. Kendall's Tau (τ)

Kendall's tau (τ) is a statistical method for determining the relationship between two permutations. It is commonly used to assess the consistency between different ranking methods. Let us consider two sequences *A* and *B*, both containing *N* elements: $A = (a_1, a_2, ..., a_N)$ and $B = (b_1, b_2, ..., b_N)$. For any pair of two elements (a_i, b_i) and (a_j, b_j) (where $i \neq j$), if $a_i > a_j$ and $b_i > b_j$ or if $a_i < a_j$ and $b_i < b_i$, these two pairs are referred to as concordant pairs. Conversely, if $a_i > a_j$ and $b_i < b_j$ or if $a_i < a_j$ and $b_i < b_i$, these two pairs are referred to as discordant pairs. Notably, if $a_i = a_j$ or $b_i = b_j$, then these two pairs are neither concordant nor discordant [46]. The formula for calculating the Kendall's tau (τ) between these two permutations is as follows:

$$\tau = \frac{2(n_+ - n_-)}{N(N - 1)} \tag{14}$$

where n_{-} and n_{+} are the numbers of discordant and concordant pairs. Kendall's tau (τ) quantifies the similarity between node ranking sequences under various comparative algorithms and DNC. A higher τ values signifies a stronger resemblance between the node ranking sequences of the two methods.

5.2.3. Monotonicity

To further assess the accuracy of the ranking algorithm, we employ the monotonicity index, denoted as *M*, to measure the monotonicity of the DNC method and compare it with the monotonicity of the comparative methods [47]. The definition of ranking monotonicity is as follows:

$$M(X) = \left[1 - \frac{\sum_{r \in V} N_r(N_r - 1)}{N(N - 1)}\right]^2$$
(15)

where *X* represents the node sequence, *r* is the ranking index, and N_r denotes the total count of nodes in the sequence with a rank of *r*. When M(X) approaches 1, this indicates a higher degree of monotonicity in the sequence, with all nodes having distinct rank values. When M(X) approaches 0, the network has only one ranking value, implying that all nodes have the same rank value. Additionally, we employed rank distribution plots to assess DNC and the baseline methods.

5.3. Experimental Results and Analysis

5.3.1. Correlation Analysis

First, we utilized the Kendall's tau (τ) correlation coefficient to assess the correlation between DNC and the baseline methods. A high correlation between DNC and the baseline methods indicates that the node importance ranking sequences produced by DNC are very similar to those of the baseline methods. This suggests that DNC may be utilizing similar information as the baseline methods to assess node importance. Figure 2 illustrates the correlation coefficients between the node importance sequences computed by DNC and the baseline methods in the Adjnoun, Celegans, Yeast, and Hamsterster networks. The correlation results for other networks are presented in Appendix A.

From the abovementioned correlation matrices of different networks, we can observe that the τ -values between the node importance rankings acquired from the CI and DNC methods are consistently high in the majority of networks, generally exceeding 0.7. This indicates a strong similarity between these two sequences. Furthermore, in most networks, the correlation coefficients between KS and LGM with DNC are also relatively high, suggesting a substantial similarity between the node importance rankings produced by DNC and these two methods. A high level of correlation is also observed among the rankings generated by the CI, KS, and LGM methods.



Figure 2. Subfigures (**a**–**d**) represent the correlation matrices between DNC and the baseline methods in the Adjnoun, Celegans, Yeast, and Hamsterster networks. The correlation coefficients between the two methods are mathematically expressed by the corresponding rows and columns. The first row represents the correlation coefficients between DNC and the baseline methods, while the second row represents the correlation coefficients between CI and the remaining methods. The black numbers in the figure indicate that the two methods have low correlation.

In summary, the node importance rankings yielded by DNC closely resemble those produced by CI, KS, and LGM. DNC likely utilizes comparable information to these metrics for assessing node importance. The SC method also exhibits a relatively high correlation with DNC in most networks, with a few exceptions. Conversely, the correlation coefficients between DNC, CC, as well as KSGC, are generally low in most networks, indicating significant differences between the node importance rankings obtained by DNC and these two methods; DNC may have incorporated more information. At the same time, the correlations among these metrics are additionally influenced by the network's underlying structure.

5.3.2. Connectivity Test

Figure 3 displays the robustness testing process of the real networks, where the collapse is simulated by successively removing the nodes of utmost importance within the network using the DNC method as well as the eight other node importance methods. In the experimental procedure, node importance values are computed using the DNC method and several other node importance identification methods; nodes are ranked based on these

values. After that, the nodes are deleted in declining order of importance. Figure 3 depicts the trends in $\sigma(i/N)$ for the Dmela, Kohonen, Vote, and Yeast networks, while the trends for the remaining networks are incorporated into Appendix B.



Figure 3. Subfigures (**a**), (**b**), (**c**), and (**d**) respectively represent the performance of DNC and the baseline methods in terms of accuracy on the Dmela, Kohonen, Yeast, and Vote networks. The x-axis of each subfigure represents the sequential removal of nodes according to DNC or the baseline methods. The y-axis measures the degree of network collapse, with a faster descent indicating greater accuracy for the method.

From the results in Figure 3 and Appendix B, it is obvious that the robustness curve for DNC, with nodes removed in the order determined by the DNC algorithm, is positioned at the bottom, resulting in the smallest area with the x-axis. This suggests that removing nodes via the DNC method hastens network breakdown. Table 4 provides specific robustness values for DNC and the comparative algorithms. In summary, DNC outperforms the baseline methods in terms of accuracy.

Network	DNC	CI	KS	CC	KSGC	LGM	LGC	SC	IGC
Dolphins	0.2862	0.2882	0.2947	0.3548	0.3005	0.3033	0.3153	0.3122	0.3124
Polbooks	0.2604	0.2669	0.3481	0.3341	0.2787	0.2796	0.3042	0.3146	0.2994
Adjnoun	0.2913	0.3025	0.3308	0.3260	0.3096	0.3050	0.3252	0.3276	0.3166
Jazz	0.4399	0.4384	0.4559	0.4201	0.4459	0.4438	0.4470	0.4463	0.4488
C_elegans	0.3311	0.3480	0.3689	0.3960	0.3534	0.3563	0.3849	0.3849	0.3646
USAir97	0.1230	0.1300	0.1546	0.1367	0.1379	0.1414	0.1508	0.1525	0.1562
Vote	0.1781	0.2188	0.2200	0.2954	0.3000	0.2169	0.2623	0.2418	0.2641
Email	0.2573	0.2702	0.2935	0.2893	0.2699	0.2710	0.2846	0.2828	0.2890
Yeast	0.2203	0.2368	0.2833	0.2500	0.2414	0.2452	0.2772	0.2704	0.2953
Hamsterster	0.1487	0.1422	0.1760	0.1605	0.1997	0.1461	0.1621	0.1604	0.1624
Kohonen	0.1085	0.1424	0.1708	0.2683	0.3908	0.1676	0.2722	0.2661	0.1674
Dmela	0.1293	0.1423	0.1681	0.1746	0.1463	0.1486	0.1789	0.1774	0.1673

Table 4. The robustness values (R) of DNC and the baseline methods across the different datasets show that, in the majority of networks, DNC exhibits the least robustness. The smallest R values in the different networks are shown in bold.

5.3.3. Monotonicity

We assessed the monotonicity of DNC compared to the other comparative methods using the M and rank distribution charts. Table 5 presents the specific results of M for DNC and the baseline methods. Figure 4 depicts the rank distribution charts for various methods on the Dmela, Email, Yeast, and Hamsterster networks. As observed in Figure 4 and Appendix C, in all rank distribution plots, the DNC method consistently forms a straight line towards the lower end, indicating the excellent monotonicity of the DNC method. From the above monotonicity index M values and rank distribution plots, it can be observed that LGC exhibits the best monotonicity, with an average value closest to 1. However, LGC has a high time complexity. The next best in terms of monotonicity are KSGC, LGM, and DNC, while K-Shell demonstrates the weakest monotonicity. LGC considers the cumulative total of the Laplacian centrality scores for nodes within half the distance of the node's truncation radius as an attraction model's mass, leading to its high time complexity. Similarly, the time complexity of KSGC and LGM based on the gravity model is also high. DNC, on the other hand, takes into account only a node's degree and its local clustering coefficient within the first-order neighbors. Unless a node's values for these two factors are identical, DNC can effectively differentiate the importance of different nodes. Furthermore, DNC has a time complexity of O(|E|) and exhibits strong monotonicity. Consequently, when considering all factors, DNC's performance is relatively superior.

Table 5. The M value of DNC and the baseline methods. The largest M values in the different networks are shown in bold.

Network	DNC	CI	KS	CC	KSGC	LGM	LGC	SC	IGC
Dolphins	0.9958	0.9613	0.3769	0.9737	0.9852	0.9821	0.9979	0.9675	0.3124
Polbooks	1.0000	0.9993	0.4949	0.9847	0.9985	0.9967	1.0000	0.9887	1.0000
Adjnoun	0.9994	0.9846	0.5990	0.9837	0.9981	0.9961	1.0000	0.9920	0.9997
Jazz	0.9993	0.9980	0.7944	0.9878	0.9994	0.9991	0.9996	0.9983	0.9994
C_elegans	0.9977	0.9949	0.6094	0.9893	0.9974	0.9972	0.9979	0.9955	0.9977
USAir97	0.9951	0.9433	0.8114	0.9892	0.9935	0.9933	0.9981	0.9928	0.9951
Vote	0.9956	0.9100	0.7265	0.9988	0.9994	0.9993	0.9999	0.9887	0.9997
Email	0.9993	0.9649	0.8088	0.9988	0.9982	0.9977	0.9999	0.9943	0.9999
Yeast	0.9856	0.9111	0.7737	0.9988	0.9988	0.9986	0.9996	0.9873	0.9992
Hamsterster	0.9819	0.9641	0.8714	0.9851	0.9848	0.9844	0.9957	0.9829	0.9857
Kohonen	0.9954	0.9332	0.7306	0.9980	0.9965	0.9960	0.9997	0.9943	0.9984
Dmela	0.9491	0.8583	0.7083	0.9996	0.9996	0.9995	0.9999	0.9905	0.9998
Average	0.9914	0.9533	0.6883	0.9908	0.9957	0.9950	0.9988	0.9879	0.9414



Figure 4. Subfigures (**a**), (**b**), (**c**), and (**d**) respectively depict the rank distribution plots of DNC and the baseline methods on the Dmela, Email, Yeast, and Hamsterster networks. The x-axis of each subfigure represents the ranking of nodes, while the y-axis represents the count of nodes with the same ranking.

5.3.4. CPU Time

The time complexity of DNC is O(|E|). However, solely comparing time complexity may have limitations in some networks. Therefore, this paper also compares the CPU running time of DNC and the baseline methods. To facilitate a better comparison of the CPU running time of DNC and other comparative algorithms across various networks, we used the Python programming language, version 3.9.0, and employed the same coding style for execution on an identical computer. Figure 5 illustrates a comparison of CPU running time for Adjnoun, Jazz, Dolphins, and USAir97. Table 6 displays the runtime comparison between DNC and the baseline methods across various networks.

The data presented in Figure 5 and Table 6 make it clear that, across all networks, the K-Shell algorithm exhibits the highest efficiency. K-Shell only requires iteratively removing nodes with a degree less than or equal to *k*, which is the most efficient operation in this situation. However, the accuracy of K-Shell is quite low. The DNC and SC algorithms also exhibit relatively high efficiency and require minimal CPU running time. This is because SC only involves a simple addition of first-degree and second-degree values. Although the CPU running time for DNC and SC is quite comparable, DNC's accuracy is significantly higher than that of K-shell and SC. Therefore, our proposed DNC algorithm combines both time efficiency and accuracy.



Figure 5. Subfigures (**a**), (**b**), (**c**), and (**d**) respectively illustrate the CPU time comparison between DNC and the baseline methods on the Adjnoun, Jazz, Dolphins, and USAir97 networks.

Network	DNC	CI	KS	CC	KSGC	LGM	LGC	SC	IGC
Dolphins	0.0008	0.0012	0.0002	0.0016	0.0136	0.0131	0.0733	0.0006	0.0126
Polbooks	0.0020	0.0039	0.0004	0.0043	0.0419	0.0412	0.3333	0.0034	0.0442
Adjnoun	0.0022	0.0054	0.0004	0.0052	0.0479	0.0469	0.2430	0.0017	0.0837
Jazz	0.0175	0.0368	0.0015	0.0359	0.3764	0.4438	1.0518	0.0063	0.3777
C_elegans	0.0110	0.0413	0.0015	0.0457	0.4396	0.4176	2.4965	0.0078	0.7293
USAir97	0.0195	0.0520	0.0050	0.1501	1.3578	1.1940	6.6021	0.0084	1.1002
Vote	0.0206	0.0964	0.0046	0.6506	5.6010	5.5481	115.3479	0.0186	4.4040
Email	0.0294	0.1170	0.0045	0.6634	5.7952	5.7732	130.6723	0.0225	6.0688
Yeast	0.0736	0.1971	0.0105	3.1377	27.2292	27.7196	1832.8288	0.4662	9.3199
Hamsterster	0.1881	1.0533	0.0157	6.7333	44.7648	45.4922	3458.4607	0.6355	27.4336
Kohonen	0.3810	3.7765	0.0425	23.3436	137.8985	139.2541	138,334.9177	0.5790	243.8257
Dmela	0.1596	3.0046	0.1210	141.5445	615.0009	597.0216	342,162.0082	0.1133	268.1242

6. Comparative Analysis of DNC Performance in Small World, Erdős–Rényi, and Scale-Free Networks

The Watts–Strogatz small world network (WS) is renowned for its high local clustering and short average path length, making it suitable for simulating social networks. The Erdős– Rényi random graph (ER) provides a simple randomly connected network structure. The Barabasi–Albert scale-free network (BA) better characterizes the power-law distribution of node degrees. By validating the DNC method in these three types of networks, we can comprehensively illustrate its performance characteristics in diverse network environments. We constructed networks for these three models, setting the number of nodes to 889 and approximately 2914 edges. We evaluated DNC's effectiveness in identifying crucial nodes based on robustness, monotonicity, and CPU time.

Figure 6 depicts the variation trends of the DNC method and baseline methods in the BA network. Figure 7 illustrates the variation trends of the DNC method and the baseline methods in the ER graph. Figure 8 showcases the variation trends of the DNC method and the baseline methods in the WS network.



Figure 6. Performance comparison of DNC and the baseline methods on the BA network.



Figure 7. Performance comparison of DNC and the baseline methods on the ER network.



Figure 8. Performance comparison of DNC and the baseline methods on the WS network.

The robustness results 'R' compared with eight other algorithms are presented in Table 7. This comparison is presented for academic accuracy.

Network	DNC	CI	KS	CC	KSGC	LGM	LGC	SC	IGC
BA	0.1973	0.2323	0.2937	0.3323	0.2937	0.2410	0.3209	0.3094	0.2467
WS	0.4385	0.4395	0.4867	0.4595	0.4831	0.4438	0.4532	0.4538	0.4562
ER	0.3771	0.3989	0.4305	0.4239	0.4390	0.3914	0.4053	0.4061	0.4157

Table 7. Robustness of DNC and the baseline methods on the BA, WS, and ER networks.

From Table 7, it can be observed that the robustness value of the DNC method is the lowest. Therefore, in the aforementioned three networks, DNC demonstrates higher accuracy compared to the baseline methods.

Figure 9 presents the rank distribution of the DNC method and baseline methods in the BA network. Figure 10 displays the rank distribution of the DNC method and baseline methods in the ER graph. Figure 11 illustrates the rank distribution plot on the WS network. From these three rank distribution graphs, it is evident that the DNC method is capable of drawing a nearly flat line at the bottom.



Figure 9. Rank distribution plots of DNC and the baseline methods on BA networks.



Figure 10. Rank distribution plots of DNC and the baseline methods on ER networks.

Table 8 displays the monotonicity values 'M' of the DNC method and the baseline methods in the WS, ER, and BA networks. While the monotonicity 'M' of the DNC method may not be the highest, it is still relatively close to 1. As mentioned earlier, DNC outperforms the baseline methods in terms of robustness. This is presented for academic accuracy.



Figure 11. Rank distribution plots of DNC and the baseline methods on WS networks.

Table 8. Monotonicity indices *M* of DNC and the baseline methods on BA, WS, and ER networks.

Network	DNC	CI	KS	CC	KSGC	LGM	LGC	SC	IGC
BA	0.9810	0.9983	0.3183	0.9972	0.9944	0.9895	1.0000	0.9762	1.0000
WS	0.9999	0.9977	0.0438	0.9956	0.9873	0.9775	1.0000	0.9641	1.0000
ER	0.9691	0.9980	0.0579	0.9973	0.9937	0.9893	1.0000	0.9751	1.0000

Subsequently, we conducted an analysis of CPU running times, as shown in Table 9. It is evident that methods based on the gravity model, while exhibiting good monotonicity, have significantly higher time complexity. For DNC and CI, in most network structures and special networks, the robustness of DNC is lower than that of CI, indicating that DNC has a higher accuracy in identifying crucial nodes in the network. Although DNC slightly lags behind CI and K-Shell in terms of CPU running time, its accuracy surpasses both CI and K-Shell, and its time complexity is reasonable.

Table 9. CPU running time of DNC and the baseline methods on BA, WS, and ER networks.

Network	DNC	CI	KS	CC	KSGC	LGM	LGC	SC	IGC
BA	0.1627083	0.0640015	0.0034087	0.3464144	2.9167356	2.9463526	55.3866124	0.0902765	3.7456719
WS	0.1311109	0.0437059	0.0031796	0.3523955	3.2523763	3.2334455	66.9728192	0.1548067	2.9645162
ER	0.1070077	0.0406134	0.0034559	0.3901652	3.4885044	3.5960308	59.9473234	0.0861778	2.3780626

In summary, DNC demonstrates excellent performance in WS, ER, and BA networks. This is presented for academic precision.

7. Conclusions

In summary, this paper introduces the DNC method, which is based on both the intrinsic characteristics of nodes and the information from their neighboring nodes. DNC computes a node's importance score as the sum of its degree and the local clustering coefficient of its first-order neighbors, so its time complexity is very low. Comparative experiments involving eight different centrality metrics on various real networks were conducted. The experimental results highlight DNC's strong performance concerning accuracy, monotonicity, and time efficiency. Nevertheless, despite these strengths, DNC exhibits certain limitations. It focuses solely on local network information and could be extended to consider global network attributes to enhance its performance. Additionally, DNC has the potential for extension to weighted and directed networks. The insights gained from this research will play a pivotal role in advancing the development of efficient techniques for identifying and protecting critical nodes within network systems.

Author Contributions: Conceptualization, N.Z. and S.Y.; methodology, S.Y.; software, J.W.; validation, S.Y., H.W. and X.Z.; formal analysis, N.Z.; investigation, J.W.; resources, T.L.; data curation, H.W.; writing—original draft preparation, S.Y.; writing—review and editing, S.Y.; visualization, N.Z.;

supervision, J.W.; project administration, T.L.; funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Key Research and Development Program of Yunnan Province (202102AA100021); the National Natural Science Foundation of China (62066048 and 62366057); the demonstration project of comprehensive government management and large-scale industrial application of the major special project of CHEOS: 89-Y50G31-9001-22/23; and the Science Foundation of Yunnan Province (202101AT070167) and supported by a grant from the Key Laboratory for Crop Production and Smart Agriculture of Yunnan Province (2022ZHNY10).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available from the corresponding author upon request. The data are not publicly available due to the following reasons. The information contained in the data is proprietary to the funding organization that supported this research, and public sharing is restricted to protect their intellectual property and competitive interests.

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

Appendix A

DNC and the baseline methods' correlation matrix on 12 networks.



Figure A1. Cont.







Figure A1. Subfigures (**a**–**l**) represent the correlation matrices between DNC and baseline methods on Adjnoun, Dolphins, Polbooks, Jazz, Celegans, USAir97, Dmela, Vote, Email, Yeast, Hamsterster, Kohonen networks, respectively.

Appendix B

DNC and the baseline methods' performance in terms of accuracy on 12 networks.



Figure A2. Cont.



Figure A2. Cont.



Figure A2. Subfigures (**a–l**) represent the accuracy performance of DNC versus baseline methods on Adjnoun, Dmela, Celegans, Email, Hamsterster, Kohonen, Jazz, Vote, Polbooks, Dolphins, Yeast, USAir97 networks, respectively.

Appendix C

DNC and the baseline methods' rank distribution charts on 12 networks.



Figure A3. Cont.



Figure A3. Subfigures (**a**–**l**) represent the rank distribution plots of DNC and baseline methods on Adjnoun, Dmela, Celegans, Dolphins, Email, Hamsterster, Jazz, Kohonen, Yeast, Polbooks, USAir97, and Vote networks, respectively.

References

- 1. Albert, R.; Barabási, A.-L. Statistical Mechanics of Complex Networks. Rev. Mod. Phys. 2002, 74, 47–97. [CrossRef]
- Tulu, M.M.; Mkiramweni, M.E.; Hou, R.; Feisso, S.; Younas, T. Influential Nodes Selection to Enhance Data Dissemination in Mobile Social Networks: A Survey. J. Netw. Comput. Appl. 2020, 169, 102768. [CrossRef]
- Ahmed, H.; Howton, T.C.; Sun, Y.; Weinberger, N.; Belkhadir, Y.; Mukhtar, M.S. Network Biology Discovers Pathogen Contact Points in Host Protein-Protein Interactomes. *Nat. Commun.* 2018, *9*, 2312. [CrossRef] [PubMed]

- Guo, S.; Zhou, D.; Fan, J.; Tong, Q.; Zhu, T.; Lv, W.; Li, D.; Havlin, S. Identifying the Most Influential Roads Based on Traffic Correlation Networks. *EPJ Data Sci.* 2019, *8*, 28. [CrossRef]
- 5. Liu, Q.; Wang, J.; Zhao, Z.; Zhao, N. Relatively Important Nodes Mining Algorithm Based on Community Detection and Biased Random Walk with Restart. *Phys. A Stat. Mech. Appl.* **2022**, *607*, 128219. [CrossRef]
- 6. Schadt, E.E. Molecular Networks as Sensors and Drivers of Common Human Diseases. *Nature* **2009**, *461*, 218–223. [CrossRef]
- 7. Boguñá, M.; Krioukov, D.; Claffy, K.C. Navigability of Complex Networks. Nat. Phys. 2009, 5, 74–80. [CrossRef]
- 8. Leskovec, J.; Adamic, L.A.; Huberman, B.A. The Dynamics of Viral Marketing. ACM Trans. Web 2007, 1, 5-es. [CrossRef]
- 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]
- 10. Zhao, N.; Wang, H.; Wen, J.; Li, J.; Jing, M.; Wang, J. Identifying Critical Nodes in Complex Networks Based on Neighborhood Information. *New J. Phys.* **2023**, *25*, 083020. [CrossRef]
- 11. Zhang, J.-X.; Chen, D.-B.; Dong, Q.; Zhao, Z.-D. Identifying a Set of Influential Spreaders in Complex Networks. *Sci. Rep.* **2016**, *6*, 27823. [CrossRef] [PubMed]
- Fan, C.; Zeng, L.; Sun, Y.; Liu, Y.-Y. Finding Key Players in Complex Networks through Deep Reinforcement Learning. *Nat. Mach. Intell.* 2020, 2, 317–324. [CrossRef]
- 13. Freeman, L.C. Centrality in Social Networks Conceptual Clarification. Soc. Netw. 1978, 1, 215–239. [CrossRef]
- 14. Bonacich, P.; Lloyd, P. Eigenvector-like Measures of Centrality for Asymmetric Relations. Soc. Netw. 2001, 23, 191–201. [CrossRef]
- 15. Sabidussi, G. The Centrality of a Graph. Psychometrika 1966, 31, 581–603. [CrossRef]
- Kitsak, M.; Gallos, L.K.; Havlin, S.; Liljeros, F.; Muchnik, L.; Stanley, H.E.; Makse, H.A. Identification of Influential Spreaders in Complex Networks. *Nat. Phys.* 2010, *6*, 888–893. [CrossRef]
- 17. Zareie, A.; Sheikhahmadi, A. A Hierarchical Approach for Influential Node Ranking in Complex Social Networks. *Expert Syst. Appl.* **2018**, *93*, 200–211. [CrossRef]
- 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. Appl.* 2020, 554, 124229. [CrossRef]
- 19. Zeng, A.; Zhang, C.-J. Ranking Spreaders by Decomposing Complex Networks. Phys. Lett. A 2013, 377, 1031–1035. [CrossRef]
- Ma, L.; Ma, C.; Zhang, H.-F.; Wang, B.-H. Identifying Influential Spreaders in Complex Networks Based on Gravity Formula. Phys. A Stat. Mech. Appl. 2016, 451, 205–212. [CrossRef]
- 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]
- Liu, F.; Wang, Z.; Deng, Y. GMM: A Generalized Mechanics Model for Identifying the Importance of Nodes in Complex Networks. *Knowl.-Based Syst.* 2020, 193, 105464. [CrossRef]
- Li, Z.; Huang, X. Identifying Influential Spreaders by Gravity Model Considering Multi-Characteristics of Nodes. *Sci. Rep.* 2022, 12, 9879. [CrossRef] [PubMed]
- Xu, G.; Dong, C. CAGM: A Communicability-Based Adaptive Gravity Model for Influential Nodes Identification in Complex Networks. *Expert Syst. Appl.* 2024, 235, 121154. [CrossRef]
- 25. Yu, E.-Y.; Wang, Y.-P.; Fu, Y.; Chen, D.-B.; Xie, M. Identifying Critical Nodes in Complex Networks via Graph Convolutional Networks. *Knowl.-Based Syst.* 2020, 198, 105893. [CrossRef]
- 26. Ou, Y.; Guo, Q.; Xing, J.-L.; Liu, J.-G. Identification of Spreading Influence Nodes via Multi-Level Structural Attributes Based on the Graph Convolutional Network. *Expert Syst. Appl.* **2022**, *203*, 117515. [CrossRef]
- 27. Zenil, H.; Kiani, N.A.; Zea, A.A.; Tegnér, J. Causal Deconvolution by Algorithmic Generative Models. *Nat. Mach. Intell.* **2019**, *1*, 58–66. [CrossRef]
- Zenil, H.; Kiani, N.A.; Marabita, F.; Deng, Y.; Elias, S.; Schmidt, A.; Ball, G.; Tegnér, J. An Algorithmic Information Calculus for Causal Discovery and Reprogramming Systems. *iScience* 2019, 19, 1160–1172. [CrossRef]
- Zenil, H.; Kiani, N.A.; Tegnér, J. Low-Algorithmic-Complexity Entropy-Deceiving Graphs. *Phys. Rev. E* 2017, 96, 012308. [CrossRef]
- Morone, F.; Makse, H.A. Influence Maximization in Complex Networks through Optimal Percolation. *Nature* 2015, 524, 65–68. [CrossRef]
- Li, Z.; Ren, T.; Ma, X.; Liu, S.; Zhang, Y.; Zhou, T. Identifying Influential Spreaders by Gravity Model. Sci. Rep. 2019, 9, 8387. [CrossRef] [PubMed]
- 32. Zhang, Q.; Shuai, B.; Lü, M. A Novel Method to Identify Influential Nodes in Complex Networks Based on Gravity Centrality. *Inf. Sci.* 2022, *618*, 98–117. [CrossRef]
- Zhou, F.; Lü, L.; Mariani, M.S. Fast Influencers in Complex Networks. Commun. Nonlinear Sci. Numer. Simul. 2019, 74, 69–83. [CrossRef]
- 34. Wang, J.; Li, C.; Xia, C. Improved Centrality Indicators to Characterize the Nodal Spreading Capability in Complex Networks. *Appl. Math. Comput.* **2018**, 334, 388–400. [CrossRef]
- 35. Watts, D.J.; Strogatz, S.H. Collective Dynamics of 'Small-World' Networks. Nature 1998, 393, 440-442. [CrossRef]
- Zachary, W.W. An Information Flow Model for Conflict and Fission in Small Groups. J. Anthropol. Res. 1977, 33, 452–473. [CrossRef]

- Lusseau, D.; Schneider, K.; Boisseau, O.J.; Haase, P.; Slooten, E.; Dawson, S.M. The Bottlenose Dolphin Community of Doubtful Sound Features a Large Proportion of Long-Lasting Associations. *Behav. Ecol. Sociobiol.* 2003, 54, 396–405. [CrossRef]
- Rossi, R.A.; Ahmed, N.K. The Network Data Repository with Interactive Graph Analytics and Visualization. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25 January 2015; AAAI Press: Austin, TX, USA; pp. 4292–4293.
- Newman, M.E.J. Finding Community Structure in Networks Using the Eigenvectors of Matrices. *Phys. Rev. E* 2006, 74, 036104. [CrossRef]
- 40. Gleiser, P.; Danon, L. Community Structure in Jazz. Advs. Complex. Syst. 2003, 06, 565–573. [CrossRef]
- Leskovec, J.; Huttenlocher, D.; Kleinberg, J. Signed Networks in Social Media. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA, 10 April 2010; Association for Computing Machinery: New York, NY, USA; pp. 1361–1370.
- 42. Bu, D.; Zhao, Y.; Cai, L.; Xue, H.; Zhu, X.; Lu, H.; Zhang, J.; Sun, S.; Ling, L.; Zhang, N.; et al. Topological Structure Analysis of the Protein-Protein Interaction Network in Budding Yeast. *Nucleic Acids Res.* **2003**, *31*, 2443–2450. [CrossRef]
- 43. Guimerà, R.; Danon, L.; Díaz-Guilera, A.; Giralt, F.; Arenas, A. Self-Similar Community Structure in a Network of Human Interactions. *Phys. Rev. E Stat. Nonlin Soft Matter Phys.* **2003**, *68*, 065103. [CrossRef] [PubMed]
- Rozemberczki, B.; Davies, R.; Sarkar, R.; Sutton, C. GEMSEC: Graph Embedding with Self Clustering. In Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Vancouver, BC, Canada, 27–30 August 2019; pp. 65–72.
- 45. Schneider, C.M.; Moreira, A.A.; Andrade, J.S.; Havlin, S.; Herrmann, H.J. Mitigation of Malicious Attacks on Networks. *Proc. Natl. Acad. Sci. USA* **2011**, *108*, 3838–3841. [CrossRef] [PubMed]
- 46. Sun, Z.; Sun, Y.; Chang, X.; Wang, F.; Wang, Q.; Ullah, A.; Shao, J. Finding Critical Nodes in a Complex Network from Information Diffusion and Matthew Effect Aggregation. *Expert Syst. Appl.* **2023**, 233, 120927. [CrossRef]
- Bae, J.; Kim, S. Identifying and Ranking Influential Spreaders in Complex Networks by Neighborhood Coreness. *Phys. A Stat. Mech. Appl.* 2014, 395, 549–559. [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.