Computing and Informatics, Vol. 40, 2021, 1422–1443, doi: 10.31577/cai_2021_6_1422

ENHANCED CRITICAL NODE DETECTION IN SOCIAL NETWORKS

Leila Ajam*, Seyed Naghi Seyedaghaee

Department of Computer Engineering, Aliabad Katoul Branch
Islamic Azad University
Aliabad Katoul, Iran
e-mail: leilaajam@yahoo.com, sn_seyedaghaee@aliabadiau.ac.ir

Abstract. In this paper, we investigate the popular centrality-based approaches to find a set of critical nodes whose deletion causes the most disconnectivity in the network. Demonstrating the weak points of these approaches which only consider a ranking factor, we propose an Enhanced Critical Node Detection (ECND) method which can work with any kind of ranking score by considering the structure of a network. We have designed a set of experiments using 24 different artificial and real-world networks, varying in the number of vertices and number of edges. Using two different objective functions including the number of connected components and the weighted average size of the connected components, experimental results show outperformance of ECND in comparison to all 8 other methods.

Keywords: Social network analysis, critical node detection, disconnectivity, centrality, social networks

Mathematics Subject Classification 2010: 91D30, 91C20, 62H30

1 INTRODUCTION

As many real-world systems can be represented by networks, a vast range of researchers from different science areas have focused on analyzing networks and studying their structural behavior [1]. Social and biological networks [2, 3, 4, 5, 6], in which vertices represent individuals or proteins and edges represent communications

* Corresponding author

or interactions, are the two most famous examples of networks that have attracted a lot of research, especially during the past two decades. Among complex network analysis approaches, Critical Node Detection (CND) is an important task which given a graph $G(V, E)$ aims to find a set of vertices $S \in V$ whose removal will leave the graph with minimal pairwise connectivity [7]. In this regard, critical nodes whose removal will break the network into more components with minimum variance in their size are more desirable. Identifying critical nodes of a network can reveal important insights about the structure and vulnerability of the network. Critical node detection can also reveal leaders in social networks and potential hubs in telecommunication or supply chain networks.

A node is important if its failure or malicious behavior essentially changes network performance [8]. Generally, the optimal solutions for different versions of CND will not be the same. Concurrently, they have a certain level of correlation [9]. Ranking vital nodes of networks is very meaningful for numerous applications, such as disease propagation inhibition and information dissemination control [10]. Various criterias for CND have been suggested, and the effectiveness of these influence measures has been investigated for the case where the complete network structure is known [11]. Widespread usage of complex interconnected social networks such as Facebook, Twitter, and LinkedIn also increased the attractiveness of CND problems [12].

Unfortunately, the critical node detection problem is known as an NP-complete problem which makes it very hard to be solved optimally even for medium size problems. Therefore, a range of approximation, evolutionary and heuristic algorithms have been proposed to tackle the critical node detection problem. However, due to a high variety of network structures, one measure or algorithm may not work well on different structures. Hence, one challenging issue in current approaches of critical node detection is robustness which means having an algorithm working accurately enough to discover critical nodes in a range of network structures. One of the main approaches for detecting critical nodes in social networks is based on centrality analysis. Centrality-based approaches (like degree, betweenness, etc) have been used to measure the relative importance of nodes in both weighted and unweighted graphs in social network analysis context [13].

However, research has shown that the deletion of nodes of high centrality might not necessarily result in maximal network disconnectivity [14, 15]. This fact demonstrates the importance of having a more sophisticated selection approach for identifying critical nodes in complex networks. To address this important issue in the critical node detection problem, this research proposes a more accurate approach for identifying a set of nodes whose deletion can lead to higher loss in network connectivity. More precisely, we improved the centrality-based critical node detection methods by taking into account the network structure in the neighborhood of the node that is a candidate for removal. Given a ranking vector of importance for each node in the network, our method starts from the highest rank and in each step assesses the nominated candidate and only adds it to the selected list of critical nodes, if it passes the assessment criterion. The proposed algorithm then updates the degrees of the neighbors of the selected node before reiterating the procedure for the next nominated candidate from the ranking list. Briefly, the main contribution of this paper is developing a robust algorithm for critical node detection which produces more reliable and stable results in different network structures. Another important contribution of this paper is reviewing eight popular centrality-based approaches for critical node detection and also implementing and using them in the experiments. Hence, we have comprehensive experimental results of the proposed ECND method and statistically comparison its performance with all of the eight well-known approaches on 24 different complex networks. The rest of this paper is organized as follows. In Section 2, we review existing centrality-based approaches for critical node detection problems and provide information for eight of the most popular centrality-based methods. Then in Section 3, we present our proposed algorithm and describe the idea behind that by an explanatory example. Then we discuss its procedure and express how it can improve the performance of the current approaches by detecting the right set of critical nodes. Section 4 empirically discusses the performance of the proposed algorithm on different types of artificial and real-world datasets and statistically compares it to all the eight discussed methods in Section 2. Finally, the conclusion of this paper is presented in Section 5.

2 RELATED WORK

The problem of critical node detection was formally introduced in 2009 [7]. For an un-weighted un-directed network $G(V, E)$, a set of nodes $S \in V$, $|S| < k$ whose deletion minimizes the network connectivity are called critical nodes. k is defined by the user and determines the maximum number of critical nodes. Mathematically, the objective of the CNDP is to determine:

$$S = \arg\min_{S \in V} \sum_{i,j \in (V \setminus S)} u_{i,j}\, G(V, S); \qquad |S| < k, \tag{1}$$

where

$$u_{i,j} = \begin{cases} 1, & \text{if exists a path between } i \text{ and } j; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

As shown in Equation (2), pairwise connectivity of a graph is calculated by summation of binary values u_{ij} for all pairs of nodes. The u_{ij} is 1 if there is a way to access j from i and 0 otherwise. Since its introduction, the critical node detection problem has attracted a lot of attention in the research society. Until now, an important approach for solving this problem relies on centrality analysis. A variety of centrality measures have been used to discover critical nodes. In most centrality-based approaches, k critical nodes are identified as a set of k nodes with the highest centrality. The most straightforward centrality measure is called degree centrality which is based on the idea that nodes with higher degrees are more important as they are connected to more network members. Crucitti et al. [16] studied the vulnerability of scale-free complex networks by attacking high degree nodes in benchmark datasets. They showed that scale-free networks are more resilient in comparison with random networks to random node removals, but they are fragile to attacks on high degree nodes. Another centrality measure frequently used for critical node detection is Betweenness centrality which measures the number of times that a node is in the shortest path between any two other nodes in a graph. The betweenness centrality of a node can be computed as follows:

$$C_B(v) = \sum_{i \neq j \neq v} \sigma_{ij}(v)/\sigma_{ij} \tag{3}$$

where σ_{ij} is the total number of shortest paths between i and j and $\sigma_{ij}(v)$ is the number of those paths which pass through v. As it is embedded in the definition of betweenness, removal of k high betweenness nodes from a network minimizes the maximum and average paths in the remaining network.

The closeness centrality measures the importance of each node in spreading information to other nodes based on the total shortest path length between that node and all other nodes. Nodes in the center of the graph have the lowest total shortest path, and therefore their closeness value is the highest. The closeness centrality of node $i \in V$ is defined as Equation (4) where d_{ij} is the length of the shortest path between nodes i and j [17].

$$C_C(i) = \frac{|V| - 1}{\sum_{j \in V,\, j \neq i} d_{ij}}. \tag{4}$$

Freeman et al. [18] studied closeness centrality to detect the central nodes on different network structures such as wheel, circle, and chain and compared it with the betweenness centrality. They showed that removing the nodes according to distance-based centralities like betweenness is more effective than closeness in terms of breaking the network into more connected components. Brin et al. [19] proposed PageRank to identify the most important webpages in the Google search engine. This method was faster and more accurate than previous engines. The PageRank algorithm ranks each node of graphs based on its degree and its neighbors rank. In undirected networks, the PageRank of each node is calculated based on the sum of PageRanks of its neighbors. The formula that calculates the PageRank of node i in an undirected graph is as follows:

$$PR(i) = \frac{1-b}{|V|} + \sum_{j \in \Gamma(I)} \frac{PR(j)}{\deg(j)} \tag{5}$$

where $PR(i)$ is the PageRank of node i, and b is the damping factor which is the probability that surfing the network would continue (this number is suggested by Brin and Page [19] to be 0.85). The nodes with the higher ranks are considered to be more important because they either have many edges or some neighbors with high ranks [19]. In spite of many distance-based centrality measures, PageRank can be computed for a very large network in a reasonable time.

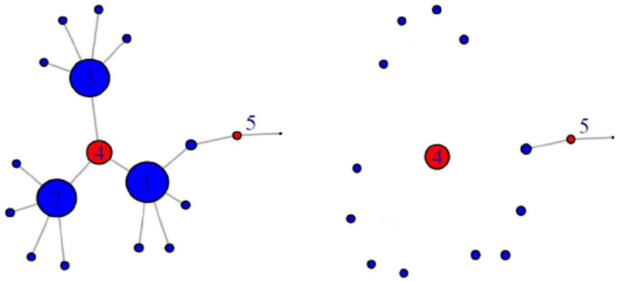
Kleinberg's Authority [20] is another measure of centrality which is defined based on the principal eigenvector of $A^T A$ where A is the adjacency matrix of the graph. For undirected graphs that the adjacency matrix is symmetric and thus $A^T A = A\,A^T$, the Authority measure has the same score as Hub (another Kleinberg's centrality measure). Eigenvector centrality [21] is another way of calculating the importance of nodes in a network which is based on the first eigenvector of the graph adjacency matrix. The eigenvector centrality of each node is proportional to the sum of the centralities of those nodes to whom it is connected. In general, vertices with high eigenvectors are those which are connected to many other vertices which are, in turn, connected to many others. This can imply that the largest values will be obtained by individuals in large cliques or high-density substructures. Another common centrality measure in the literature is Bonacich's alpha centrality [22]. Alpha centrality of the vertices in a graph is defined as the solution of the following matrix equation:

$$x = \alpha A^T x + e \tag{6}$$

where A is the (not necessarily symmetric) adjacency matrix of the graph, e is the vector of exogenous sources of the status of the vertices and α is the relative importance of endogenous versus exogenous factors. Power centrality [23] implies that a node's centrality is equal to a function of the centrality of those they are connected to. Thus, nodes which are tied to very central nodes should have higher centrality than those which are not. There is a parameter that determines the radius of the impact that centrality of each node can affect others' centrality. Small values of this parameter limit the effects to close neighbors and larger values have a global impact. According to this centrality measure, it is also possible that nodes have a negative impact on the centrality power of their neighbors. The last centrality-based critical node detection algorithm that we used in our experiments is subgraph centrality [24], which measures the number of subgraphs that a vertex is a member of them. The subgraph centrality of a vertex is defined as the number of closed loops starting at the vertex, where longer loops have exponentially smaller weights. The weak point of this method is its requirement to all eigenvalues and eigenvectors of graph adjacency matrix, which is time-consuming and makes usage of this algorithm to be limited from small to middle-sized graphs in practice. All above-mentioned centrality-based critical node detection algorithms assign a score to each node of a given network and in the next step select k nodes of the highest ranks as k critical ones. Although each of those k discrete nodes has a high score of that specific objective function, the set of k highest ones of them may not be the best set of k that could optimize its objective value. That was the key motivation for us to think more about the second phase which led us to a wise selection strategy. The proposed selection method is further discussed in detail in the next section.

3 PROPOSED METHOD

Given a centrality measure for a network, this paper proposes a new method for critical node detection problem which performs more robust in different network structures. The proposed method, called Enhanced Critical Node Detection (ECND), can work based on any ranking measure. To clarify the mechanism of the proposed method, we first explain typical approaches for CNDP.



a) An example network, the size of each node indicates centrality score

b) After deletion of first three nodes with highest centrality scores

Figure 1. An explanatory toy example

Figure 1 a) shows a network with 18 vertices. The size of each node in this figure is proportional to its ranking (centrality score) which means the bigger the node the better position in the ranking. Suppose that we want to solve a 4-critical node detection in this network ($k = 4$). In the first three steps, the centrality-based critical node detection algorithms remove the first three nodes with the biggest centrality values. Node number 4 is the next candidate to be removed as it has the fourth greatest rank. But as it is pictured in Figure 1 b), by removing node number 4, not only we do not increase disconnectivity of the network but also decrease the number of connected components because we remove an individual component. In this example, we can clearly see that the best choice for removal as a critical node in the network of Figure 1 b) is node number 5 whose ranking/centrality score was even smaller than node number 4. This simple network is an insightful example of the motivation behind the proposed ECND approach. It shows that the optimal set of k critical nodes is not essentially a set of k nodes with the highest ranking. It implies that although centrality-based ranking is a key approach for detecting vital nodes in complex networks, it should be combined with a systematic selection strategy to have the most possible impact and efficiency.

The two basic principles behind the proposed algorithm are:

1. Importance of centrality analysis for critical node detection.
2. Wise deletion based on given centrality scores.

This paper wants to highlight that nodes with high centrality scores are important to be considered as critical nodes, but the centrality should not be the only parameter for a critical node detection algorithm. Because centrality, in general, indicates the state of being accessible from other nodes especially neighbors and it slightly differs from the objective of the critical node deletion problem which is selecting a set of k nodes from a graph that maximizes disconnectivity in the remaining network. Thus, the different functionality of centrality requires a critical node detection algorithm to be wise in the selection of nodes for removal from the network. In the proposed critical node detection method we establish a wise selection strategy that takes into account both the centrality value of the nodes and the relative importance of their neighbors. In the example of Figure 1 a), node 4 is no longer a critical node when all of its three neighbors are more eligible to be a critical node than itself. The ECND algorithm looks at both structures of the graph and the ranking score of nodes simultaneously to get improve the performance of detecting the right nodes to be removed from the network, so that the disconnectivity of the network will be maximized. Given any kind of node ranking in an input network, the ECND starts from a node with the highest rank and adds it to the list of selected nodes, if its degree is at least two; which means that this node is not either a single isolated node or an end node (leaf). Because in each of these two conditions, removing the node does not increase the disconnectivety of the graph. After adding a node to a set of selected nodes, the ECND algorithm updates degrees of its neighbors (directly connected nodes to the selected node) by decreasing 1. Then, it picks the second-ranked node in the network based on its given ranking and this procedure will iterate until it reaches the desired number of k. This simple, however effective selection way, guarantees that the disconnectivity objective improves monotonically. The ECND algorithm is also shown in Algorithm 1.

4 EXPERIMENTAL RESULTS

In this section, we present an empirical analysis of our critical node detection algorithm and compare it with some of the most popular CND methods in the literature.

4.1 Datasets

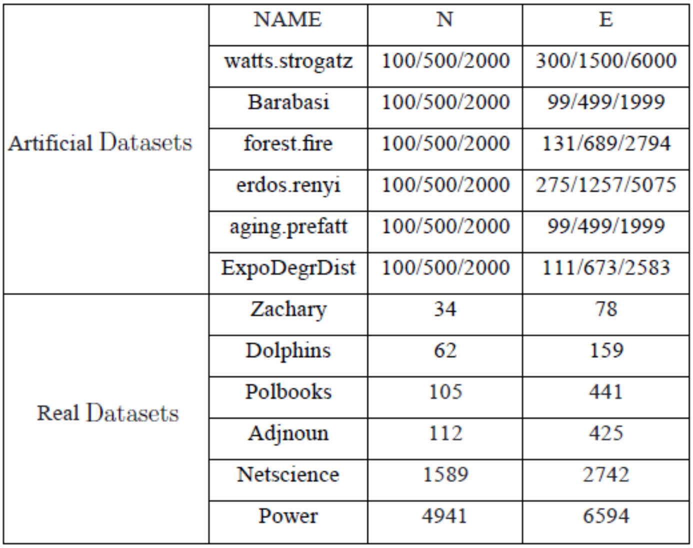
To present a comprehensive comparison, we generated 6 different categories of artificial benchmark models and 6 real-world datasets that are popular in the literature.

Table 1 briefly represents some information about the datasets used in our experiments. Artificial datasets are generated in three different sizes of 100, 500, and 2 000 nodes, as shown in the third column of this table. The number of edges of each network is also reported in the fourth column. The second part of Table 1 presents names, number of nodes, and number of edges of real datasets. More details about the benchmark models and real datasets are as follows:

Algorithm 1 The Pseudocode of the proposed ECND algorithm

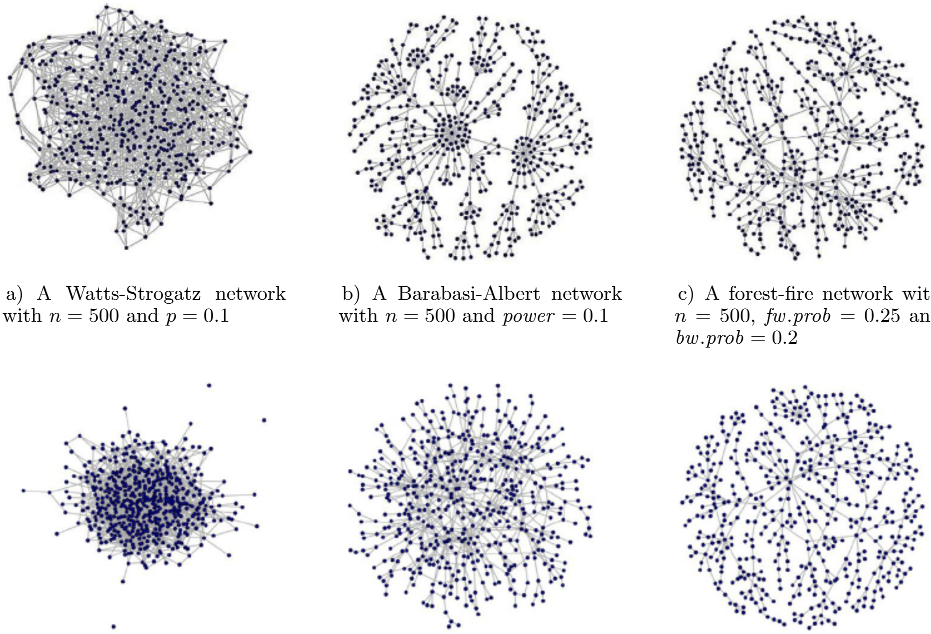
```
Input: input network $G$, ranking scores $R$, number of the critical node $k$,
Output $\Theta$: set of $k$ critical nodes to be removed from the network,
$\Theta = \emptyset$                                  // list of selected nodes
$X$ = all nodes in $G$                              // list of not assessed nodes
While (length ($\Theta$) $< k$ and $X <> \emptyset$) do
   candidate = node with highest $R$ score $\in x$
   If deg(candidate) >= 2) then
       Add candidate to $\Theta$
       $\Gamma_{\text{candidate}}$ = neighbor(candidate)
       for all $ni \in \Gamma_{\text{candidate}}$ do
           deg($ni$) = deg($ni$) $-$ 1          // update neighbors
       end for
    end if
    Remove candidate from $X$
end while
Return $\Theta$
```

- The **Watts-Strogatz** model [25] aims to generate networks that have characteristics observed in real small-world networks. The small-world phenomenon is one of the common properties of real complex networks. Two aspects of small-world networks are the low diameter of the network compared to its size and its high clustering coefficient. Watts-Strogatz model starts with a ring of n vertices



	NAME	N	E
Artificial Datasets	watts.strogatz	100/500/2000	300/1500/6000
	Barabasi	100/500/2000	99/499/1999
	forest.fire	100/500/2000	131/689/2794
	erdos.renyi	100/500/2000	275/1257/5075
	aging.prefatt	100/500/2000	99/499/1999
	ExpoDegrDist	100/500/2000	111/673/2583
Real Datasets	Zachary	34	78
	Dolphins	62	159
	Polbooks	105	441
	Adjnoun	112	425
	Netscience	1589	2742
	Power	4941	6594

Table 1. Datasets



a) A Watts-Strogatz network with $n = 500$ and $p = 0.1$

b) A Barabasi-Albert network with $n = 500$ and $power = 0.1$

c) A forest-fire network with $n = 500$, $fw.prob = 0.25$ and $bw.prob = 0.2$

d) A Erdös-Rényi network with $n = 500$ and $power = 0.01$

e) Aging-prefatt network with $n = 500$, $pa.exp = 1$ and $aging.exp = -1$

f) Exponential network with $n = 500$ and $power = -i/2$

Figure 2. Artificial benchmark networks

and connects each vertex in the ring to all of its k nearest neighbors. Then, each edge is chosen with probability p to reconnect one of its endpoints to a randomly chosen node. Figure 2 a) shows an example of the Watts-Strogatz network with 500 nodes and $p = 0.1$.

- The **Barabasi-Albert** model [26] proposes that graph structure is a result of two processes, growth and preferential attachment. Generating a network starts with m_0 nodes and grows step by step. At each step, one node is added. The last added node is linked to m existing nodes by some edges according to the probability of:

$$p(e_{new,v} \in E) = \frac{\deg(v)}{\sum_w \deg(w)} \tag{7}$$

where $\deg(v)$ returns the degree of a vertex and v, w are the vertices that already existed in the graph. As a result of the preferential attachment characteristic of this model which is conducted by the probability of Equation (7), nodes already have high degrees are more likely to grow further and increase their degrees. The power-law edge distribution can make this model a good representative of real-world networks. Figure 2 b) shows an example of such a network with 500 nodes with *power* = 1. As it was expected, there are many nodes with small degree and a few ones with the highest degree in this network.

- Another network that is used in our experiments is called the **Forest-Fire** network model [27]. The Forest Fire network generating algorithm starts with a random single node $v \in V$. Then, it picks an existing node w with uniform distribution and creates the edge (v, w). Then, it randomly generates two numbers x and y with binomial distribution through $\frac{p}{1-p}$ and $\frac{rp}{1-rp}$, where p and $r \in (0, 1)$ are called forward and backward probabilities, respectively. In the next phase, the algorithm creates an edge from v to each of $x + y$ vertices and will repeat this process recursively from those $x + y$ vertices. Figure 2 c) shows an example of such a network with 500 nodes and a forward probability of 0.25 and a backward probability of 0.2. For more information about this network, you can refer to [27].
- The **Erdös-Rényi** model [28] generates a random graph in which every possible edge is created with the same constant probability p. Therefore, the number of edges in graph $G(n, p)$ is a random variable with the expected value $\binom{n}{2}p$. Figure 2 d) shows an example of the Erdös-Rényi network with $n = 500$ and $p = 0.01$.
- The **Aging-Prefatt** is a discrete-time step model of a growing graph. It starts with a single node with no edge and then in each time step, a new vertex is added and it initiates several edges to the old vertices in the network. The probability that an old vertex is connected to is:

$$p[i] \sim (c \cdot k_i^\alpha + \alpha)(d \cdot l_i^\beta + b) \tag{8}$$

  where k_i is the degree of vertex i in the current time step and l_i is the age of vertex i. The age is simply defined as the number of time steps passed since the vertex is added. An example of such a network with 500 nodes and preferential attachment of 1 and an aging exponential of -1 is presented in Figure 2 e).
- The **Exponential** degree sequence network generates a network that has an exponential degree sequence with *power* $= -i/2$ where i is the index of each node. An example of an exponential degree distributed network of size 500 is shown in Figure 2 f).
- **Real Datasets:** Zachary Karate Club [29] is a social network of friendships between 34 members of a karate club at a US university in the 1970s. Dolphin dataset [30] is a social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand. Political Book is a network of books about US politics published around the time of the 2004 presidential election and sold by the online bookseller Amazon.com. Edges between books represent frequent co-purchasing of books by the same buyers. Adjacent nouns [31] is an adjacency network of common adjectives and nouns in the novel David Copperfield by Charles Dickens. The Network science dataset [31] is

a coauthorship network of scientists working on network theory and experiment. The last dataset is called the power dataset [25] and represents the topology of the Western States Power Grid of the United States.

4.2 Comparison Measure

According to the definition of the critical node detection problem, the discovered k critical nodes are expected to maximize (pairwise) disconnectivity of the network. Here we consider two different criteria to assess disconnectivity of the resulting networks after deletion of critical nodes detected by different approaches.

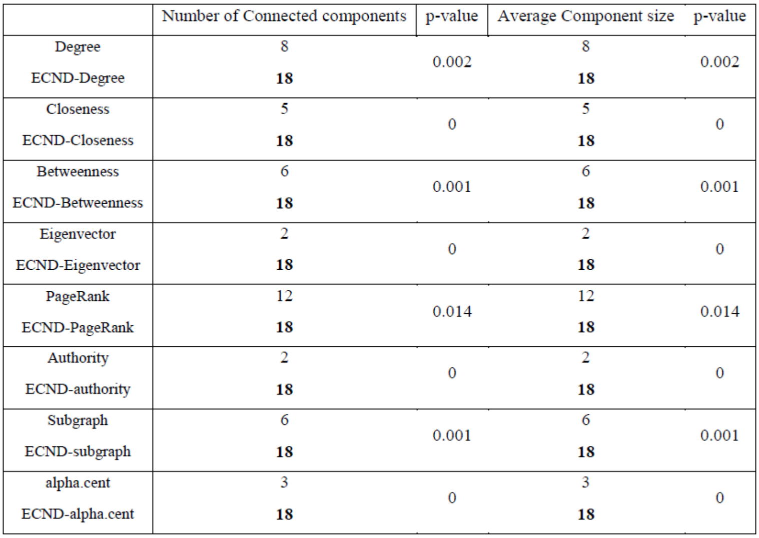
- Number of network's connected components after deletion of critical nodes is our first measure of pairwise disconnectivity. This criterion is a common measure in the literature for analyzing the effect of deleting critical nodes and comparing the results of different approaches. Clearly, the better set of critical nodes is the one whose deletion results in a more disconnected network with a higher number of connected components.
- Weighted average of component size is also an important criterion for evaluating the importance of identifying critical nodes. Critical nodes are supposed to break down network structures into pieces of uniformly sized (balanced) components. For instance, a set of 4 critical nodes whose deletion breaks down a network of size 100 to two almost equal size components are much more preferred to those who break the network to the component size of 5 and 95. Hence we are interested to find a set of critical nodes with a smaller weighted average component size. The weighted average of two components sizes of $x_1 = 1$ and $x_2 = 9$ is $(\frac{x_1}{x_1+x_2}x_1) + (\frac{x_2}{x_1+x_2}x_2) = (0.1 \times 1) + (0.9 \times 9) = 8.2$ while for two component sizes of $x_1 = x_2 = 5$ it is equal to $(0.5 \times 5) + (0.5 \times 5) = 5$.

4.3 Comparison with Other Methods

In this section, we apply our enhanced critical node detection framework on a range of centrality measures and illustrate results on different datasets. We also present a sensitivity analysis of the numbers of critical nodes.

Test on artificial benchmarks. Table 2 represents the results of experiments on artificial datasets. In this experiment, we used 6 artificial benchmark models which are explained in the previous section and generated 3 datasets of sizes 100, 500, 2 000 based on each model. Hence, in total, this experiment is done on 18 datasets of 6 benchmark models and 3 different sizes. For this experiment, the number of critical nodes is set to $k = 0.15 \times N$, where N is the dataset size. To summarize the results, instead of reporting all 18 values for 18 datasets, we report the number of datasets (out of a total of 18) that each method achieved the best result. The first column of Table 2 represents a list of methods in pairs grouping each centrality-based method with its corresponding ECND approach.

In the second column, the number of wins for each method is reported. Note that a win for the first criterion is equivalent to higher number of connected components and for the second one represents a smaller average component size. Note that if two methods have the same results, both of them are considered winners. The result of the statistical pairwise Friedman test is reported in the last column of Table 2. Null hypothesis of this test is two methods have no significant difference in performance. The null hypothesis is strongly rejected due to very small p-values which proves that the ECND statistically outperforms the corresponding centrality measure significantly.

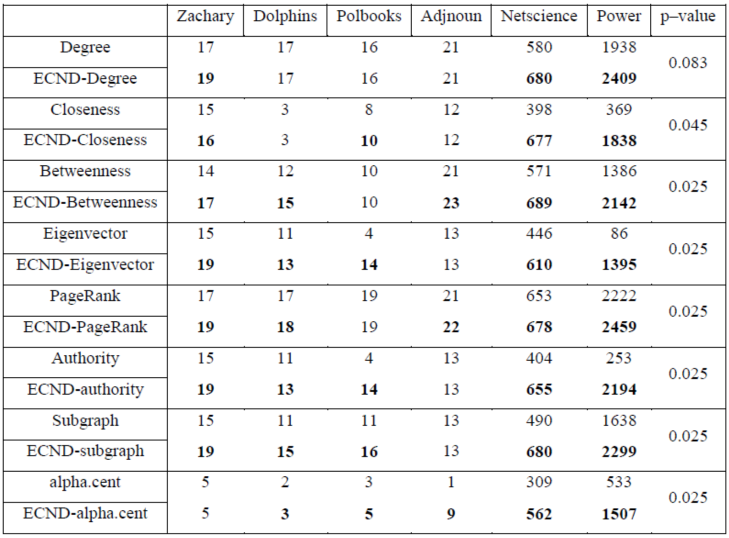


	Number of Connected components	p-value	Average Component size	p-value
Degree	8	0.002	8	0.002
ECND-Degree	**18**		**18**	
Closeness	5	0	5	0
ECND-Closeness	**18**		**18**	
Betweenness	6	0.001	6	0.001
ECND-Betweenness	**18**		**18**	
Eigenvector	2	0	2	0
ECND-Eigenvector	**18**		**18**	
PageRank	12	0.014	12	0.014
ECND-PageRank	**18**		**18**	
Authority	2	0	2	0
ECND-authority	**18**		**18**	
Subgraph	6	0.001	6	0.001
ECND-subgraph	**18**		**18**	
alpha.cent	3	0	3	0
ECND-alpha.cent	**18**		**18**	

Table 2. Statistical comparison on 18 artificial benchmarks based on two criteria

To investigate the effect of k on the performance of the proposed method in comparison with competitor algorithms, we considered the problem of k critical node detection for 8 different values of k. Ranges of k are slightly different for different datasets based on the structure of networks and their connectivity. We set range of k in [5–60] % and [5–70] % in Erdös-Rényi and Watts-Strogatz, respectively; and kept it under 30 % for the other benchmarks, due to their lower connectivity. This experiment has been done over all 6 artificial datasets of size $n = 2\,000$ and the number of connected components is used as an evaluation measure. Figure 3 demonstrates the results of this experiment for all 8 critical node detection methods and ECND. As shown in this figure, ECND improves the performance of all other methods. More precisely, when the number of critical nodes grows especially to more than 10–15 percent of network size, the difference between ECND and other methods increases remarkably. The greatest difference is 400 % in the Forest-Fire dataset when the Alpha-Centrality algorithm cannot depart the network into more than 200 components, while ECND based on the same ranking vector separates the network into more than 800 different pieces.

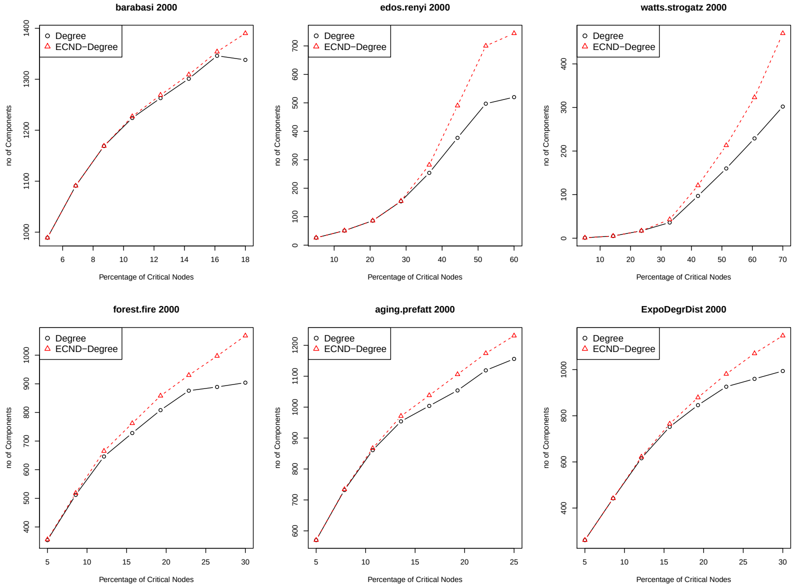
Comparing the y-axis of different figures on the same dataset in Figures 3 a), 3 b), 3 c), 3 d), 3 e), 3 f), 3 g) and 3 h) reveals that the best performing algorithm among all 8 methods in achieving the highest amount of objective value is PageRank (Figure 3 g)). Focusing on Figure 3 g) one can observe that by increasing the number of k, even the PageRank algorithm falls into trap of deleting vertices whose neighbors are also present in the set of k critical nodes; and hence are removed in advance. However, ECND can identify this trap and remain safe in such situations so that it achieves better results than PageRank.



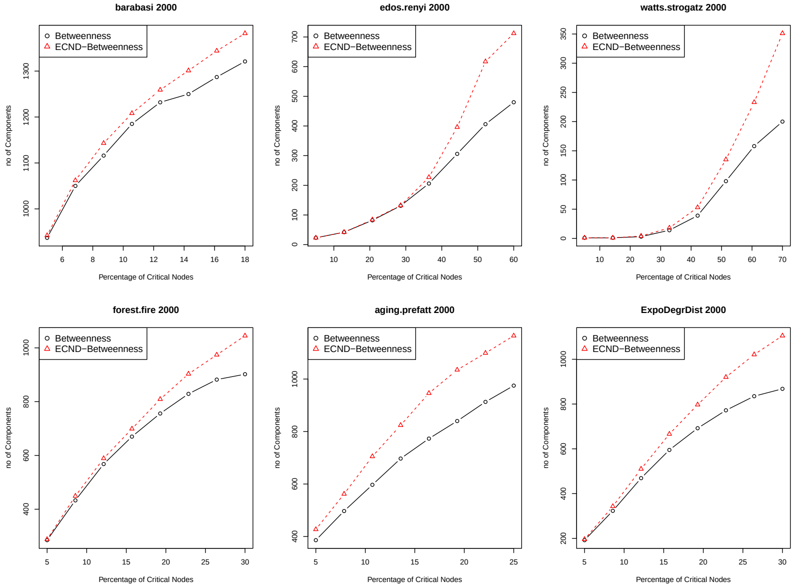
	Zachary	Dolphins	Polbooks	Adjnoun	Netscience	Power	p–value
Degree	17	17	16	21	580	1938	0.083
ECND-Degree	**19**	17	16	21	**680**	**2409**	
Closeness	15	3	8	12	398	369	0.045
ECND-Closeness	**16**	3	**10**	12	**677**	**1838**	
Betweenness	14	12	10	21	571	1386	0.025
ECND-Betweenness	**17**	**15**	10	**23**	**689**	**2142**	
Eigenvector	15	11	4	13	446	86	0.025
ECND-Eigenvector	**19**	**13**	**14**	13	**610**	**1395**	
PageRank	17	17	19	21	653	2222	0.025
ECND-PageRank	**19**	**18**	19	**22**	**678**	**2459**	
Authority	15	11	4	13	404	253	0.025
ECND-authority	**19**	**13**	**14**	13	**655**	**2194**	
Subgraph	15	11	11	13	490	1638	0.025
ECND-subgraph	**19**	**15**	**16**	13	**680**	**2299**	
alpha.cent	5	2	3	1	309	533	0.025
ECND-alpha.cent	5	**3**	**5**	**9**	**562**	**1507**	

Table 3. Test results on real datasets

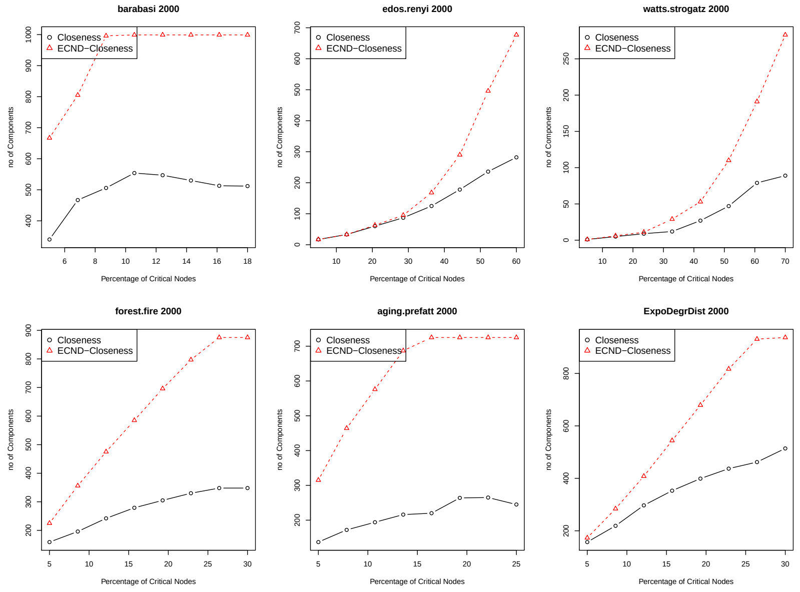
Test on real benchmarks. Tables 3 and 4 represent statistical results of ECND and other approaches on real-world networks in terms of two evaluation criteria, i.e. number of components and weighted average of components size. The first column of this table lists the different methods which are divided into pair based on each centrality measure. In each division, the first row is a typical critical node detection approach and the second row is the corresponding ECND approach based on the same centrality measure. In this experiment, k is fixed and equal to $k = 0 : 3 \times N$, where N is size of dataset. The value of evaluation criteria is reported in columns 2 to 6 of the table for each dataset. Table 3 reports results based on the number of connected components after the deletion of critical nodes. Table 4 shows the value of weighted average component size in such networks. As explained in advance, a better set of critical nodes breaks the network to a higher number of connected components which have



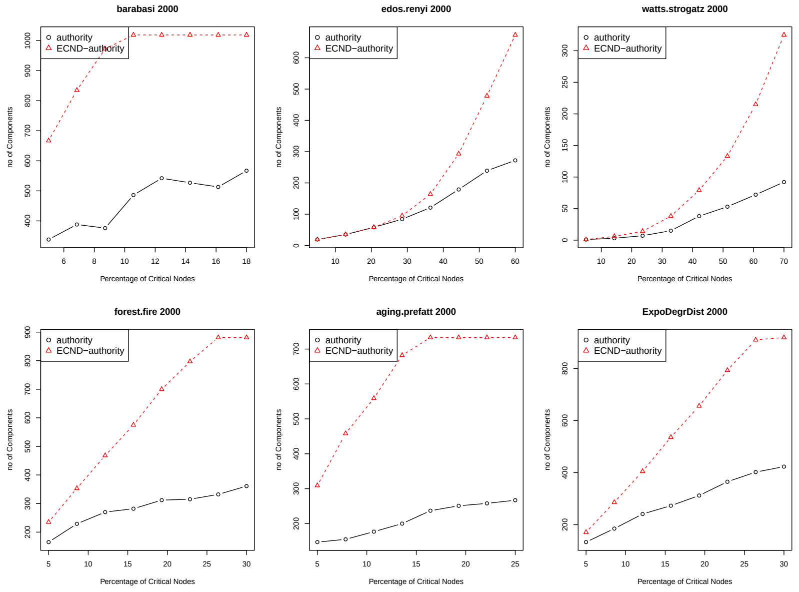
a) Effect of changing k on the performance of Degree vs. ECND-Degree on all 6 different artificial datasets



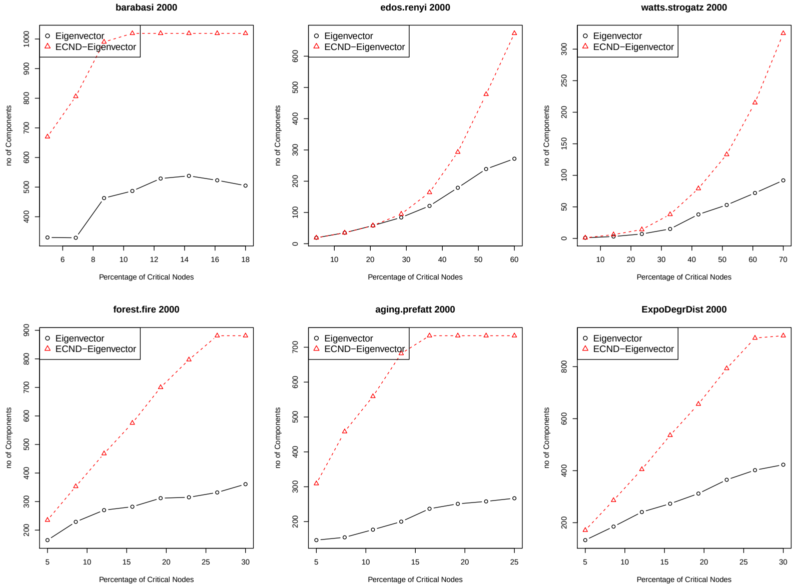
b) Effect of changing k on the performance of Betweenness vs. ECND-Betweenness on all 6 different artificial datasets



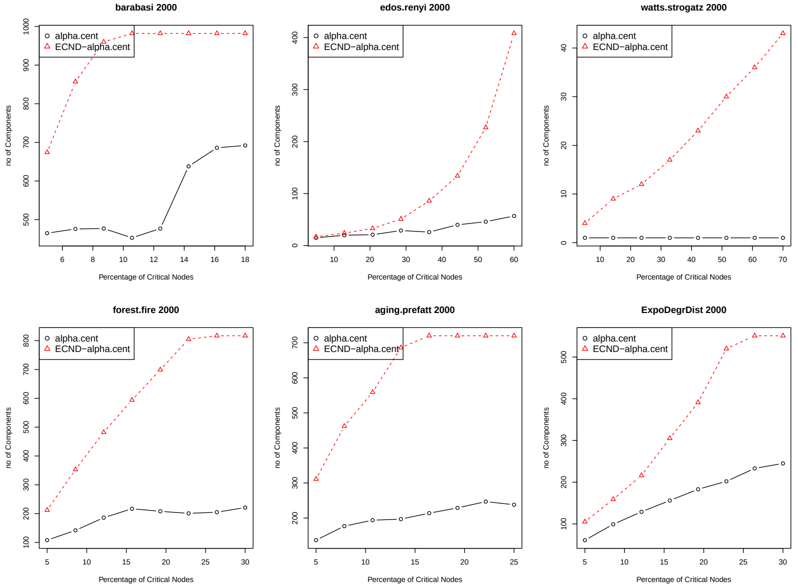
c) Effect of changing k on the performance of closeness vs. ECND-closeness on all 6 different artificial datasets



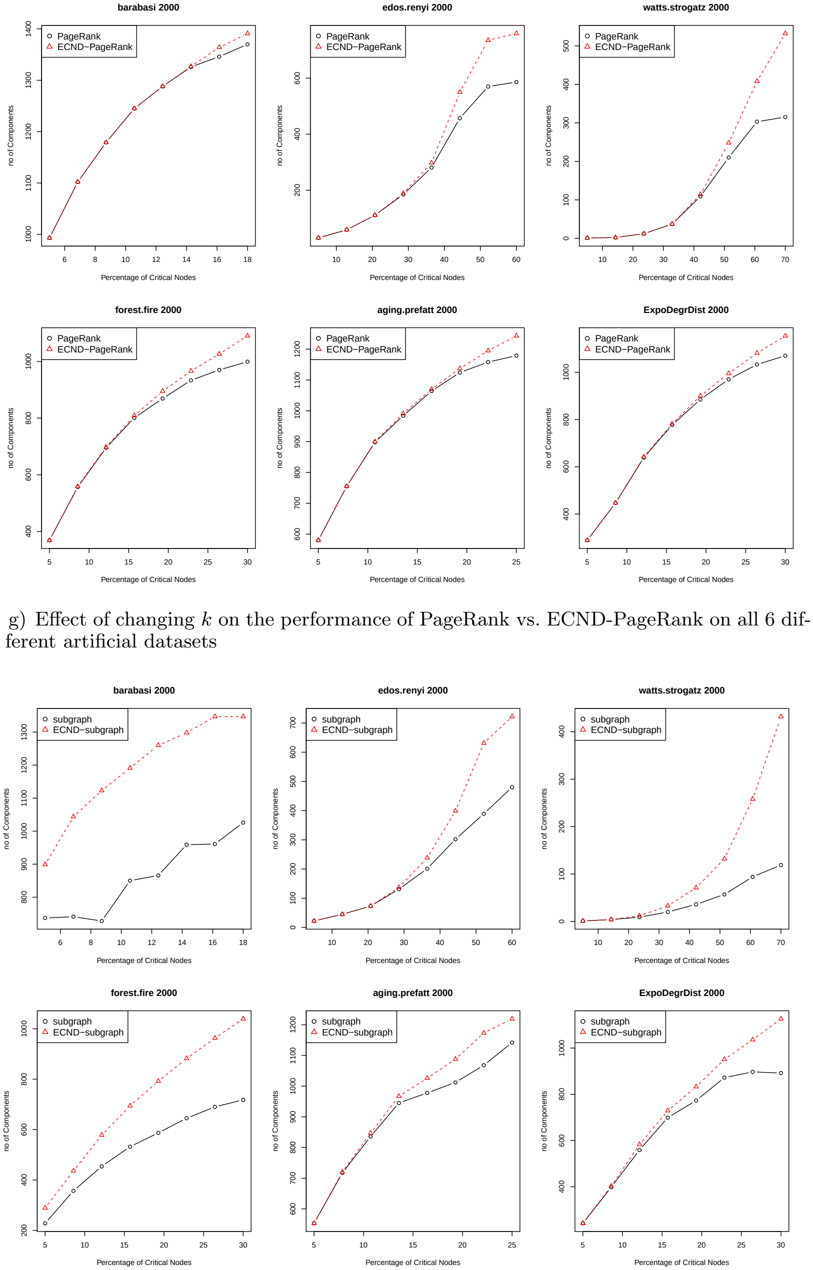
d) Effect of changing k on the performance of Authority vs. ECND-Authority on all 6 different artificial datasets



e) Effect of changing k on the performance of Eigenvector vs. ECND-Eigenvector on all 6 different artificial datasets



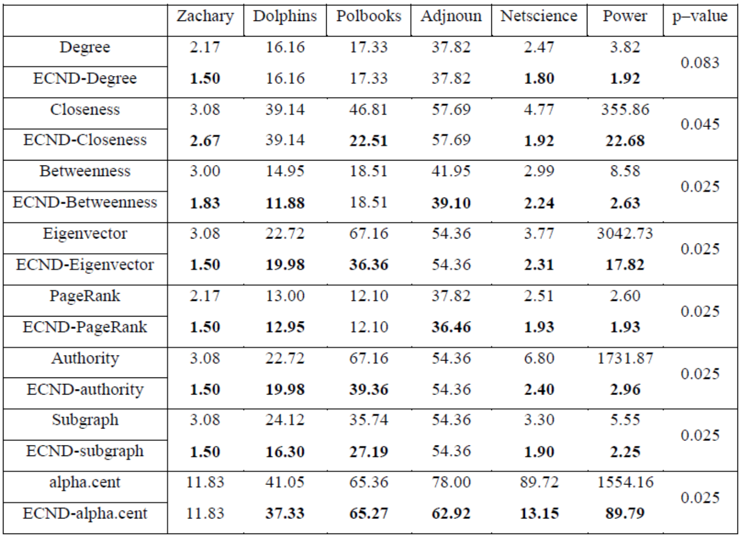
f) Effect of changing k on the performance of Alphacent vs. ECND-Alphacent on all 6 different artificial datasets



g) Effect of changing k on the performance of PageRank vs. ECND-PageRank on all 6 different artificial datasets

h) Effect of changing k on the performance of Subgraph vs. ECND-Subgraph on all 6 different artificial datasets

Figure 3.



	Zachary	Dolphins	Polbooks	Adjnoun	Netscience	Power	p–value
Degree	2.17	16.16	17.33	37.82	2.47	3.82	0.083
ECND-Degree	**1.50**	16.16	17.33	37.82	**1.80**	**1.92**	
Closeness	3.08	39.14	46.81	57.69	4.77	355.86	0.045
ECND-Closeness	**2.67**	39.14	**22.51**	57.69	**1.92**	**22.68**	
Betweenness	3.00	14.95	18.51	41.95	2.99	8.58	0.025
ECND-Betweenness	**1.83**	**11.88**	18.51	**39.10**	**2.24**	**2.63**	
Eigenvector	3.08	22.72	67.16	54.36	3.77	3042.73	0.025
ECND-Eigenvector	**1.50**	**19.98**	**36.36**	54.36	**2.31**	**17.82**	
PageRank	2.17	13.00	12.10	37.82	2.51	2.60	0.025
ECND-PageRank	**1.50**	**12.95**	12.10	**36.46**	**1.93**	**1.93**	
Authority	3.08	22.72	67.16	54.36	6.80	1731.87	0.025
ECND-authority	**1.50**	**19.98**	**39.36**	54.36	**2.40**	**2.96**	
Subgraph	3.08	24.12	35.74	54.36	3.30	5.55	0.025
ECND-subgraph	**1.50**	**16.30**	**27.19**	54.36	**1.90**	**2.25**	
alpha.cent	11.83	41.05	65.36	78.00	89.72	1554.16	0.025
ECND-alpha.cent	11.83	**37.33**	**65.27**	**62.92**	**13.15**	**89.79**	

Table 4. Test results on real datasets based on weighted average component size

a smaller average component size. Accordingly, better results are highlighted in bold in each pair of rows. As shown in Tables 3 and 4, without any exception, the ECND performed either equally or in most times better than the other corresponding centrality-based approaches. We observe in these tables that in some cases, the ECND method dramatically improves the results. For instance, on the Power dataset, the number of critical nodes identified by the ECND-Closeness breaks the network into a more than five times greater number of components (500 % improvement) than that of the corresponding method. Similarly, the ECND-Authority performs 8 times (800 % improvement) and in the ECND-Eigenvector 16 times (1 600 % improvement) more accurately than the corresponding method. In the last column of Tables 3 and 4, the results of the statistical Friedman test are reported. The null hypothesis of this test is the two methods have no significant difference in performance. The hypothesis is strongly rejected in 7 cases due to very small p-values. Table 4 confirms the results in Table 3. As previously discussed, the weighted average of components indicates how much the components are overheated in size. For instance, given $k = 2$, apparently the objective of the critical node detection problem is not breaking the input network into two pieces of 1 and $N - 2$. The best solution can be two components of almost $N/2$. Table 4 shows that not only the proposed ECND algorithm outperforms the other algorithms in the number of components, but also the quality of the results, i.e. the weighted average of connected components, is much better than that of the other methods. For example, in the Power benchmark, not only our method is performed 16 times better than the Eigenvector algorithm in aspect of number of components, but also the quality of ECND's results are more than 170 times better than that of the Eigenvector method. The similar results on the other datasets also confirm the efficiency of the proposed method.

5 CONCLUSION

In this research, we considered an algorithm called ECND for critical node detection in complex networks. ECND improves the performance of centrality-based critical node detection algorithms by taking into account both the ranking score of vertices and the structure of neighbors of candidate nodes. Experimental results on 18 artificial networks in different sizes from $n = 100$ to $n = 2\,000$ showed that the proposed ECND method significantly outperforms all other 8 well-known critical node detection algorithms. The sensitivity analysis results of the algorithm based on different ranges of k revealed that the disconnectivity of a network can dramatically increase by ECND while the other competitor algorithms converged to a steady-state. Promising results of applying the ECND algorithm on 6 real-world famous networks varied in size from $n = 34$ to $n = 4\,941$ also confirmed the outperformance of the proposed method in comparison with all other methods. We also statistically investigated the output quality of different methods in our experiments. The high quality of ECND results was another evidence for the well-performing of ECND in comparison with the other 8 existing methods. We intend to accelerate CND algorithms in our future work.

Acknowledgment

This study is extracted from the research project supported by the Islamic Azad University of Aliabad Katoul Branch.

REFERENCES

[1] Ganji, M.—Seifi, A.—Alizadeh, H.—Bailey, J.—Stuckey, P. J.: Generalized Modularity for Community Detection. In: Appice, A., Rodrigues, P., Santos Costa, V., Gama, J., Jorge, A., Soares, C. (Eds.): Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2015). Springer, Cham, Lecture Notes in Computer Science, Vol. 9285, 2015, pp. 655–670, doi: 10.1007/978-3-319-23525-7_40.

[2] Petridis, N. E.—Petridis, K.—Stiakakis, E.: Global e-Waste Trade Network Analysis. Resources, Conservation and Recycling, Vol. 158, 2020, Art. No. 104742, doi: 10.1016/j.resconrec.2020.104742.

[3] Alinejad-Rokny, H.: Proposing on Optimized Homolographic Motif Mining Strategy Based on Parallel Computing for Complex Biological Networks. Journal of Medical Imaging and Health Informatics, Vol. 6, 2016, No. 2, pp. 416–424, doi: 10.1166/jmihi.2016.1707.

[4] Alinejad-Rokny, H.—Pourshaban, H.—Orimi, A. G.—Baboli, M. M.: Network Motifs Detection Strategies and Using for Bioinformatic Networks. Journal of Bionanoscience, Vol. 8, 2014, No. 5, pp. 353–359, doi: 10.1166/jbns.2014.1245.

[5] Hosseinpoor, M.—Parvin, H.—Nejatian, S.—Rezaie, V.—Bagherifard, K.—Dehzangi, A.—Beheshti, A.—Alinejad-Rokny, H.: Proposing a Novel Community Detection Approach to Identify Cointeracting Genomic Regions. Mathematical Biosciences and Engineering, Vol. 17, 2020, No. 3, pp. 2193–2217, doi: 10.3934/mbe.2020117.

[6] Ahmadinia, M.—Meybodi, M. R.—Esnaashari, M.—Alinejad-Rokny, H.: Energy-Efficient and Multi-Stage Clustering Algorithm in Wireless Sensor Networks Using Cellular Learning Automata. IETE Journal of Research, Vol. 59, 2013, No. 6, pp. 774–782, doi: 10.4103/0377-2063.126958.

[7] Arulselvan, A.—Commander, C. W.—Elefteriadou, L.—Pardalos, P. M.: Detecting Critical Nodes in Sparse Graphs. Computers and Operations Research, Vol. 36, 2009, No. 7, pp. 2193–2200, doi: 10.1016/j.cor.2008.08.016.

[8] Lalou, M.—Tahraoui, M. A.—Kheddouci, H.: The Critical Node Detection Problem in Networks: A Survey. Computer Science Review, Vol. 28, 2018, pp. 92–117, doi: 10.1016/j.cosrev.2018.02.002.

[9] Aringhieri, R.—Grosso, A.—Hosteins, P.—Scatamacchia, R.: A General Evolutionary Framework for Different Classes of Critical Node Problems. Engineering Applications of Artificial Intelligence, Vol. 55, 2016, pp. 128–145, doi: 10.1016/j.engappai.2016.06.010.

[10] Wang, Z.—Du, C.—Fan, J.—Xing, Y.: Ranking Influential Nodes in Social Networks Based on Node Position and Neighborhood. Neurocomputing, Vol. 260, 2017, pp. 466–477, doi: 10.1016/j.neucom.2017.04.064.

[11] Tsugawa, S.—Kimura, K.: Identifying Influencers from Sampled Social Networks. Physica A: Statistical Mechanics and Its Applications, Vol. 507, 2018, pp. 294–303, doi: 10.1016/j.physa.2018.05.105.

[12] DasGupta, B.—Mobasheri, N.—Yero, I. G.: On Analyzing and Evaluating Privacy Measures for Social Networks under Active Attack. Information Sciences, Vol. 473, 2019, pp. 87–100, doi: 10.1016/J.INS.2018.09.023.

[13] Opsahl, T.—Agneessens, F.—Skvoretz, J.: Node Centrality in Weighted Networks: Generalizing Degree and Shortest Paths. Social Networks, Vol. 32, 2010, No. 3, pp. 245–251, doi: 10.1016/j.socnet.2010.03.006.

[14] Borgatti, S. P.: Identifying Sets of Key Players in a Social Network. Computational and Mathematical Organization Theory, Vol. 12, 2006, No. 1, pp. 21–34, doi: 10.1007/s10588-006-7084-x.

[15] Zhang, D.—Sterbenz, J. P. G.: Modelling Critical Node Attacks in MANETs. In: Elmenreich, W., Dressler, F., Loreto, V. (Eds.): Self-Organizing Systems (IWSOS 2013). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 8221, 2014, pp. 127–138, doi: 10.1007/978-3-642-54140-7_11.

[16] Crucitti, P.—Latora, V.—Marchiori, M.—Rapisarda, A.: Efficiency of Scale-Free Networks: Error and Attack Tolerance. Physica A: Statistical Mechanics and Its Applications, Vol. 320, 2003, pp. 622–642, doi: 10.1016/S0378-4371(02)01545-5.

[17] Zhuge, H.—Zhang, J.: Topological Centrality and Its e-Science Applications. Journal of the American Society for Information Science and Technology, Vol. 61, 2010, No. 9, pp. 1824–1841, doi: 10.1002/asi.21353.

[18] Freeman, L. C.—Roeder, D.—Mulholland, R. R.: Centrality in Social Networks: II. Experimental Results. Social Networks, Vol. 2, 1980, No. 2, pp. 119–141, doi: 10.1016/0378-8733(79)90002-9.

[19] Brin, S.—Page, L.: Reprint of: The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks, Vol. 56, 2012, No. 18, pp. 3825–3833, doi: 10.1016/j.comnet.2012.10.007.

[20] Kleinberg, J. M.: Authoritative Sources in a Hyperlinked Environment. Journal of the ACM (JACM), Vol. 46, 1999, No. 5, pp. 604–632, doi: 10.1145/324133.324140.

[21] Bonacich, P.: Power and Centrality: A Family of Measures. American Journal of Sociology, Vol. 92, 1987, No. 5, pp. 1170–1182.

[22] Bonacich, P.—Lloyd, P.: Eigenvector-Like Measures of Centrality for Asymmetric Relations. Social Networks, Vol. 23, 2001, No. 3, pp. 191–201, doi: 10.1016/S0378-8733(01)00038-7.

[23] Bonacich, P.: Factoring and Weighting Approaches to Status Scores and Clique Identification. Journal of Mathematical Sociology, Vol. 2, 1972, No. 1, pp. 113–120, doi: 10.1080/0022250X.1972.9989806.

[24] Estrada, E.—Rodríguez-Velázquez, J. A.: Subgraph Centrality in Complex Networks. Physical Review E, Vol. 71, 2005, No. 5, Art. No. 056103, doi: 10.1103/PhysRevE.71.056103.

[25] Watts, D. J.—Strogatz, S. H.: Collective Dynamics of 'Small-World' Networks. Nature, Vol. 393, 1998, No. 6684, pp. 440–442, doi: 10.1038/30918.

[26] Barabási, A. L.—Albert, R.: Emergence of Scaling in Random Networks. Science, Vol. 286, 1999, No. 5439, pp. 509–512, doi: 10.1126/science.286.5439.509.

[27] Leskovec, J.—Kleinberg, J.—Faloutsos, C.: Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (KDD '05), ACM, 2005, pp. 177–187, doi: 10.1145/1081870.1081893.

[28] Erdös, P.—Rényi, A.: On Random Graphs. Publicationes Mathematicae Debrecen, Vol. 6, 1959, pp. 290–297.

[29] Zachary, W. W.: An Information Flow Model for Conflict and Fission in Small Groups. Journal of Anthropological Research, Vol. 33, 1977, No. 4, pp. 452–473, doi: 10.1086/jar.33.4.3629752.

[30] 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. Behavioral Ecology and Sociobiology, Vol. 54, 2003, No. 4, pp. 396–405, doi: 10.1007/s00265-003-0651-y.

[31] Newman, M. E. J.: Finding Community Structure in Networks Using the Eigenvectors of Matrices. Physical Review E, Vol. 74, 2006, No. 3, Art. No. 036104, doi: 10.1103/PhysRevE.74.036104.

Leila Ajam is currently pursuing her Ph.D. degree in the Department of Computer Engineering, Gorgan Branch, Islamic Azad University, Gorgan, Iran. She also received her M.Sc. degree in computer engineering from Babol Branch, Islamic Azad University, Babol, Iran. Her research interests include data mining, data analysis, cluster ensemble, community detection.

Seyed Naghi Seyedaghaee is currently pursuing his Ph.D. degree in the Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran. He also received his M.Sc. degree in computer engineering and artificial intelligence from Mashhad Branch, Islamic Azad University, Mashhad, Iran, in 2009. He has published several papers in AI fields. His research interests include data mining, data analysis, cluster ensemble, community detection and optimization methods.