mathematics

MDPI

Article

Network Analysis Based on Important Node Selection and Community Detection

Attila Mester, Andrei Pop, Bogdan-Eduard-Mădălin Mursa, Horea Greblă, Laura Dioşan and Camelia Chira *

Department of Computer Science, Babeş-Bolyai University, 400084 Cluj-Napoca, Romania; attila.mester@ubbcluj.ro (A.M.); andrei.pop@ubbcluj.ro (A.P.); bogdan.mursa@ubbcluj.ro (B.-E.-M.M.); horea.grebla@ubbcluj.ro (H.G.); laura.diosan@ubbcluj.ro (L.D.)
* Correspondence: camelia.chira@ubbcluj.ro

Abstract: The stability and robustness of a complex network can be significantly improved by determining important nodes and by analyzing their tendency to group into clusters. Several centrality measures for evaluating the importance of a node in a complex network exist in the literature, each one focusing on a different perspective. Community detection algorithms can be used to determine clusters of nodes based on the network structure. This paper shows by empirical means that node importance can be evaluated by a dual perspective—by combining the traditional centrality measures regarding the whole network as one unit, and by analyzing the node clusters yielded by community detection. Not only do these approaches offer overlapping results but also complementary information regarding the top important nodes. To confirm this mechanism, we performed experiments for synthetic and real-world networks and the results indicate the interesting relation between important nodes on community and network level.

Keywords: network analysis; important nodes; community detection

Citation: Mester, A.; Pop, A.; Mursa, B.-E.-M.; Grebla, H.; Dioşan, L.; Chira, C. Network Analysis Based on Important Node Selection and Community Detection. *Mathematics* **2021**, *9*, 2294. https://doi.org/10.3390/math9182294

Academic Editors: Oliviu Matei, Alfonso Niño and Anca Andreica

Received: 22 July 2021
Accepted: 14 September 2021
Published: 17 September 2021

1. Introduction

The highly dense literature and studies on complex networks [1–4] demonstrate the importance of this field, which can be explained by its versatile usage, ability to model a wide variety of real systems and applicability in a series of domains such as travel, commerce, biology, technology, sociology, chemistry, economics and many other fields. Regarding a whole network, its dynamism and properties are encoded in its nodes and their wiring diagram, i.e., the links. There are a series of metrics aimed to measure these basic elements and the nodes, according to different criteria. Looking at a higher level, these nodes will always form some kind of groups and clusters whose properties are just as significant as the nodes, regarding the nature of the network.

One important research direction related to network science refers to the detection of important nodes. Several centrality metrics that can be engaged for this task are defined in the literature [3] while many studies continue to propose new measures and models to attempt better approaches [5–8]. The identification of important nodes in networks is a challenging but essential task with many high-impact applications in the real world. For example, in case of a virus, it can help identify the key sources of spreading to prevent a possible epidemic. In a large-scale network, it is compelling to back up the servers in order not to lose important data. Identifying the important nodes will increase the endurance and robustness of the network. Another example can be extracted from the series of large blackouts that happened in different countries over the last decades. A significant one is the power blackout that affected most of northern and eastern India on 30 and 31 July 2012. The one from 30th July affected over 400 million people, being the largest power outage considering all the people affected, surpassing the blackout from January 2001 in Northern India—230 million people were affected then. Such high-impact situations could have been prevented if the key nodes would have been identified and

better protected. The importance of nodes in the network can be determined by using centrality measures that have been widely used but also have some drawbacks. Each measure can work well for a certain case, but at the same time, each measure fails to obtain other structural characteristics of the network. The traditional centrality measures identify a node as being the "most central" one, but they focus on information from different perspectives. For example, if we are doing an analysis on the most popular nodes in a social network, prioritizing by betweenness centrality could be wrong; however, the same betweenness centrality measure used in the analysis of epidemics is the one that tells us which nodes should be immunized first and it seemed to be the most reliable one in this case. In other words, each measure attempts to capture the nodes' importance from a certain perspective.

A community is a subset of nodes that interact with each other more than the nodes outside that particular community [9]. The nodes within a community are more interconnected, with more edges between themselves and fewer edges with other nodes outside the community. Community detection is known to be a clustering problem and many algorithms for the detection of communities have been proposed in the literature [10]. The tendency of a node is to cluster under the law of common interests, a dynamic that leads to creating structures with specific properties.

In this paper, we investigate the important nodes in different networks by aligning the results of centrality measures with community detection methods—analyzing cases where network-level important nodes detected by different centrality measures are hubs of communities, and vice versa. The aim of our study is to determine if there is any relation between the various node centrality metrics and the community detection-based hub-dominant nodes. The main contributions of the paper are as follows: (i) a comprehensive overview on the literature of node centrality metrics and community detection algorithms; (ii) several experiments on some real world networks in order to confirm that nodes having top centrality measures overlap with those obtained by the means of community detection. Analysis of results is based on centrality scores obtained by six well-known measures (i.e., degree centrality, betweeneess centrality, closeness centrality, percolation centrality, eigen centrality and page rank), computation of hub dominance in relation to communities detected and use of the susceptible-infected-recovered model to verify the information propagation in the network. Computational experiments confirm that centrality measures and community detection yield common important nodes.

The paper is structured as follows. Section 2 presents, in a formal manner, definitions of centrality criteria. Community detection methods for finding important nodes are reviewed in Section 3. Section 4 describes the details of our approach and the flow of the experiments, while Section 5 presents the computational experiments, datasets used and the results of the analysis. Finally, we draw our conclusions in Section 6, formulating possible future directions as well.

2. Node Importance Measures

The identification of central nodes in complex networks has an important theoretical significance for the structure, propagation and synchronization of complex networks. It has a very practical value for understanding the communication, control of information and transactions between nodes. The node's importance has a philosophical implication quantified as various centrality properties. All these properties have in common that they reveal the nodes with a significant impact on the information flow of the network—so-called central nodes, e.g., in a financial context, these central nodes can be large banks or large economic entities that proxy a significant number of transactions.

Degree centrality (the ratio between total number of neighbors of a node and the number of nodes in the graph [11]) assumes that if a node has a great number of adjacent nodes, it is very influential. This can be considered to be a measurement of importance, but it only considers the information locally, not taking into account the whole global network structure. Closeness centrality [12] is expressed by the inverse of the average length of the

shortest paths between a node and all other ones in the network. Hence, the greater priority a node has, the closer it is to all different nodes in the network. It can also be considered a measurement of how easily the information passes from a node to others; a drawback of this method can be the disconnected components. Betweenness centrality [13] is expressed by the fraction of shortest paths between any two nodes that pass through this node. It atones for the deficiencies that degree and closeness centrality have. Similar to degree centrality, eigen centrality (EC) [14] is a measure for the influence of a node in a network based on the number of its neighbors. This centrality goes a step further by also taking into account how well connected a node is and how many links their connections have. By doing so, this algorithm can identify influential nodes throughout the whole network, not just for those directly connected to it. Another metric is page rank (PR) [15], derived from eigen centrality, also assigning nodes a score based on the number of total connections. The difference is that page rank algorithm also takes into consideration link direction and weight. Going further, K-shell decomposition [16] (the method where the nodes can be divided on the basis of the number of its degree—nodes with degree 1 in one bucket, etc.) has been used in quite a number of application as a tool for understanding the importance of nodes within a network structure. K-shell decomposition can be visualized as a way to understand the "goodness" of a node as distributor in a large-scale complex network. Roughly speaking, the bigger the k-shell index of a node is, the better such a node can act as a distributor in the network. The above described notions are summarized in Table 1.

Table 1. Overview of node centrality metrics.

Metric	Abbreviation	Ref.	Description
Degree centrality	DC	[11]	Number of adjacent nodes, relative to the total nodes.
Betweenness centrality	BC	[13]	Number of shortest paths passing through the node.
Closeness centrality	CC	[12]	Inverse of avg. distance to other nodes.
Eigen centrality	EC	[14]	Relative degree centrality influenced by the score of the neighbors
Page rank	PR	[15]	Like EC, but accounts for directed networks too
K-shell decomposition	KS	[16]	Reveals core-periphery structure by organizing the nodes into buckets.

3. Community Detection Methods

In the network science literature, there are various definitions regarding the concept of community, all of them centered around the idea that nodes inside a community have larger number of connections than with the outside network [17]. Thus, community means a group of nodes (a set of nodes), with the following properties: they form a connected subgraph (there exists a path from each node to every other node); nodes in community A have common properties, based on some particular similarity measurements (topical and topological features); nodes in community A have more common properties with other nodes in A than with those found in community B—the interior link occurs with higher probability compared to an outer link. Consequently, the problem of community detection consists in finding a partition of the node set N, resulting in a set of communities: $C = \{C_1, C_2, ... \,|\, C_i = \{N_i | N_i \in N\}, C_i \cap C_j = \emptyset(*), \bigcup C_i = N\}$ ($(*)$ criteria is not always required, i.e., in case of overlapping communities).

Defining the community structure of the network means in fact to determine a partition of the nodes. Since the the number of such partitions grows exponentially in the number of nodes, the literature of community detection is focused on the complexity of the algorithms applied. A recent survey paper [18] offers a valuable overview on the field of community detection. According to [18], the complexity of detecting communities depends on the nature of communities (static, dynamic, overlapping), as well as the network's topology (weighted, and/or directed graphs). One could consider the problem as a clustering of the

nodes, based on similarity measures in the network [19]; or it may be worth trying to ignore the edge direction as well [20]; or one could focus on determining leader nodes based on centrality measures and build the communities around the leaders [21]. We summarize the state-of-the-art methods in community detection.

Generally, very little information is known in advance about a network's community structure. Thus, a traditional method is to assume that the network has hierarchical topology (such as a social network for example), and determine a multilevel division of the network—in this case, hierarchical clustering methods can be used. Hierarchical clustering is based on a similarity matrix, containing node pair similarities. We can find and group together similar nodes in two different manners: join the nodes (agglomerative) or separate them (divisive). The agglomerative, bottom-up approach starts with all nodes being in a separate community, and continues to merge the most similar communities, based on the cluster-similarity measure, i.e., linkage type (single-, complete-, average-linkage). One such example is the Ravasz algorithm [22], with a complexity of $\mathcal{O}(n^2)$—similarity matrix calculation for hierarchical methods is expensive, in general. The divisive, top-down approach works in a reverse manner: it starts with all nodes being in one community, and repeatedly removes the highest centrality links until no link remains. Since the links are removed according to highest centrality, it is expected that high centrality links connect nodes from different communities, low centrality ones link nodes in the same group. This can be accomplished with several metrics: link betweenness centrality; random walk betweenness—the probability that the link $l_{i,j}$ was passed by a random walker in a path from n_i to n_j. Since this metric is computationally expensive ($\mathcal{O}(n^3)$, [2]), link betweenness is preferred as divisive method. A representative divisive method is the one proposed by Girvan and Newman in 2002 [23]. The algorithm uses the edge betweenness centrality measure to repeatedly remove the highest centrality links. The algorithm's complexity depends on the centrality measure, which is $\mathcal{O}(ln)$ (l denoting the number of edges). Removing all the links means an iteration over the links, resulting in $\mathcal{O}(l^2n)$—for sparse ($l \approx n$) network: $\mathcal{O}(n^3)$.

Another approach is to use an agent, which performs random walk in the network. One such algorithm is the Walktrap, which essentially is a agglomerative method, building on the node similarity matrix based on random walks. Since a similarity matrix update is needed in every iteration, the complexity is $\mathcal{O}((l+n)^2n)$, or $\mathcal{O}(n^3)$ in sparse networks [2,24]. Another key example is the Infomap [25], exploiting the idea that a random walker tends to be blocked into a group of densely connected nodes (community), before it exits to another group of nodes. The task is to minimize the length of the Huffman encoded path of the walker by assigning the groups a prefix and reusing the codes for the nodes within the communities. Similar to another algorithm (i.e., Louvain), nodes are repeatedly moved from one community to another, based on a target cost function—while Louvain uses modularity; Infomap uses the length of the Huffman code of the random path. The complexity is $\mathcal{O}(l\ log\ l)$ [2].

Another traditionally accepted approach is to maximize the basic community evaluation metric, the partition's modularity. Communities' modularity is an intuitive and popular metric, which can measure how well the nodes are grouped together. Some algorithms are based on the idea to maximize the overall modularity of the communities. The FastGreedy algorithm builds on the maximum modularity hypothesis: the setup that results in the highest modularity score is the optimal community structure of the network. The algorithm starts with each node forming its own community. In each iteration, for each community pair, the change in modularity is calculated if the two communities were merged (resulting an agglomerative method). The communities that result in the highest improvement of modularity are merged. The complexity is $\mathcal{O}((l+n)n)$ [2]. Clauset–Newman–Moore (CNM) [26] applies the same logic as FastGreedy, but with some special data structures achieves complexity of $O(ldlogn)$, d—the depth of the dendrogram. The Louvain algorithm [27] is a preferable option for large networks, due to its $\mathcal{O}(l)$ complexity. It starts with different node being its own commu-

nity, and the concept is to place a node n_i to one of its neighboring node's community, in a way to maximize the modularity change. Depending on the implementation details (https://sites.google.com/site/findcommunities/, accessed on 12 July 2021), the complexity is at most $\mathcal{O}(llogl)$ (or $\mathcal{O}(nlogn)$ for sparse networks) [2]. Probabilistic approaches have been proved to be successful in optimizing the modularity of the network [17,28–30].

The label propagation algorithm (LPA) was first published in 2007 [31], announcing an efficient community detection method with complexity of $\mathcal{O}(ldlogn)$, d meaning the depth of the dendrogram. The method starts with each node forming its own community. In every iteration, each node takes the label of their neighbors, based on majority vote. The paper compares the synchronous and asynchronous method of node label updating. In order to avoid oscillation of labels and ensure the termination of the algorithm, the authors propose asynchronous label update—one node at a time. The algorithm stops when no more node label update is needed—each node having the majority vote of its neighbors. Other papers improved the accuracy of the initial LPA by applying heuristic on the propagation, based on local edge betweenness [32,33]. The article from 2019 [33] uses this improved method on weighted graphs, calculating local edge betweenness [34] to prioritize the propagation of labels, reaching a complexity of $\mathcal{O}(n(\frac{l}{n})^h)$ (using $h = 2$ depth edge betweenness), meaning near linearity in the nodes for sparse networks. Edge betweenness is used as a heuristic to prioritize the selection of low score edges—the concept is that high edge betweenness links tend link nodes of different communities.

All of the algorithms mentioned above are applicable for simple graphs, some of them—with necessary adjustments—capable of solving the community detection problem for weighted and/or directed networks. Detailed benchmark results are described in [10], a valuable article from which one key point is that all modularity-based approaches can be extended to weighted and/or directed networks as well, by modifying the formula of modularity [17]. In the case of LPA, link weight and directionality can also be taken into consideration for weighting the vote of each neighboring node, depending on the implementation and practical use case. Random walk methods can be also adapted to directed networks. For example, Infomap can be applied to directed and weighted networks too [10].

It is important to note that community detection algorithms are not all deterministic. In fact, any method based on a random walker or probabilistic optimization are nondeterministic, e.g., Infomap, WalkTrap, Louvain and LPA; however, the GN method or its improved version CNM, are deterministic algorithms. The algorithm's complexity is also a key factor determining their application during the experiments.

Regarding the evaluation of the partitions, along the strictly topological properties, the quality of a partition can be also measured by considering the ground true labels of the nodes (i.e., which community they belong to). For this purpose, metrics from information theory can be used (Shannon entropy, mutual information, homogeneity, completeness). Modularity of a partition [35] is a ratio in $[-1, 1]$ [36], measuring the difference between the density of links of the community, compared to the expected density of a randomized network's partition. Coverage and performance of partition [17] measures how dense the intra-community edges are, and how well the communities separate non-linked nodes. Embeddedness of node is ratio in $[0, 1]$, measuring how many of the node's neighbors are in the same community [37]. Its formula, $e_i = \frac{k_i^{in}}{k_i}$, underlines this metric importance as node ranking method within a community. Hub dominance of community C is a ratio in $[0, 1]$, measuring the existence of a hub node, connecting to as many other nodes in C as possible [37]. Its formula, $h_C = \frac{max(k^{in})}{|C|-1}$, suggests that this metric can be used as a node importance criteria after the community structure of the network is determined. Each node has a hub dominance score within a community, this number showing the extent to which this node represents the community—the node having the maximum of these scores will be the hub-dominant node of the community.

4. Proposed Analysis of the Important Nodes

This paper investigates the important nodes in a network by combining insights offered by centrality measures and community detection methods. After we determine the ideal partitioning of the nodes according to different community detection algorithms, we match the previously calculated centrality measures on the hub-dominant nodes of each community. We want to investigate that some of these metrics yield the highest score across the entire network, in the case of these hub nodes.

Our experimental analysis aims to respond to a list of research questions. These research questions serve as a guidelines for the reader in understanding and following the study's goals with respect to methodology steps. Considering these aspects, we formulate the following research questions.

- (RQ1) Analysis of important nodes through centrality metrics: what is the best way to find a network's important nodes?
- (RQ2) Analysis of important nodes from the perspective of the community structure of the network: do the community-based node metrics yield useful node ranking?
- (RQ3) Is there any correlation between the various approaches of node centrality metrics and community-based node importance?

The first two research questions target the analysis of a network from two independent standpoints. The third research question, however, is the most important aspect of this study, since it investigates a possible improvement of the node importance ranking strategies by discovering a relation between the above mentioned methods.

In order to address RQ1, we apply the aforementioned centrality algorithms: degree, closeness, betweenness, eigenvector, percolation, page rank centralities and k-shell decomposition in order to find the most important nodes on the networks that we are studying. In order to address RQ2, we analyze networks by determining their community structure with the following community detection algorithms: Greedy modularity-based algorithm [26]; LPA [31]; Louvain [27]; Infomap [25]; Infomap', which is initialized with the result partition of LPA. Since we lack the ground truth partitioning of the nodes in the datasets, we focus on strictly topological evaluation in our experiments—calculating the most intuitive metrics regarding the accuracy of community detection: modularity, coverage and performance. Further, we compare the number of detected communities, and their sizes. Furthermore, we analyze the community structure by determining the hub-dominant nodes, and the embeddedness of the nodes—focusing on their distribution.

The proposed approach to address RQ3 is to collect the important nodes obtained by community detection, and examine whether they are also important nodes according to some node centrality criteria. We executed these steps on some real, public networks, and on some generated graphs as well, according to the existing network benchmarks.

5. Experiments and Results

This section presents the experiments for several real-world, publicly available networks in order to analyze the important nodes. The following abbreviations are used for the measures computed: DC—Degree Centrality, BC—Betweeneess Centrality, CC—Closeness Centrality, PC—Percolation Centrality, EC—Eigen Centrality, PR—Page Rank, HD—Hub Dominance.

5.1. Datasets Used

Table 2 presents the networks (http://networkrepository.com, accessed on 12 July 2021) used for our experiments: Karate club network, Dolphin network, Politicians' Facebook pages network and Google+ social network. The motivation for the selection of these networks is to offer analysis on different sized datasets. Furthermore, they are frequently used in the specialized literature.

Table 2. Attributes of the networks used in the experiments.

	Nodes	Edges	Density	Min. Degree	Max. Degree	Average Degree
Karate Club Network	34	78	0.139037	1	17	4
Dolphin Network	62	159	0.084082	1	12	5
Politicians Network	5.9 k	41.7 k	0.002390	1	323	14
Google+ Social Network	211.2 k	1.5 m	6.749×10^{-5}	1	2 k	14

During the first part of our experiments, we aim to inspect the results of the node centrality metrics and community detection algorithms for the real-world networks presented in Table 2. Then, we confirm our intuitions gained through these experiments by running the experiments on randomly generated networks, respecting the Lancichinetti–Fortunato–Radicchi (LFR) benchmark [38] and the Erdős–Rényi (ER) network model [39].

5.2. Analysis Based on Centrality Scores

We show the first five nodes, according to their centrality ranking, in Tables 3 and 4—all of these nodes are in the K-shell core layer (K-shell is a method for dividing the nodes into two big categories: core—periphery; the ones in the core layer can be considered the most important ones). We have chosen to present only the first five nodes for each network due to the fact it is easier to see the correlation between the nodes of each centrality.

In the Karate network, we can observe that the majority of our methods found nodes 1, 34 and 3 as having the best centrality score. When analyzing the Dolphin network, a slightly bigger one, we see that again a couple of nodes are the ones with best scores: 15, 37 and 38. The Politicians network seems to have a better diversity when it comes to the top 5 nodes. This aspect is due to the fact that the network is way larger than the first two. The last network used in the experiments comes to strengthen this point of view. As a conclusion, we can observe that in small networks, high centrality score nodes are repeating, while for larger networks, it no longer happens and diversity starts to appear.

Table 3. Rank of the nodes from Karate and Dolphin networks according to the centrality scores.

	Karate Network						Dolphin Network					
Ranking/Centrality	**DC**	**BC**	**CC**	**PC**	**EC**	**PR**	**DC**	**BC**	**CC**	**PC**	**EC**	**PR**
First node	34	1	1	1	34	34	15	37	37	37	15	15
Second node	1	34	3	34	1	1	38	2	41	2	38	18
Third node	33	33	34	33	3	33	46	41	38	41	46	52
Fourth node	3	3	32	3	33	3	52	38	21	38	34	58
Fifth node	2	32	9	32	2	2	34	8	15	8	51	38

Table 4. Rank of the nodes from the Politicians and Google+ networks according to the centrality scores.

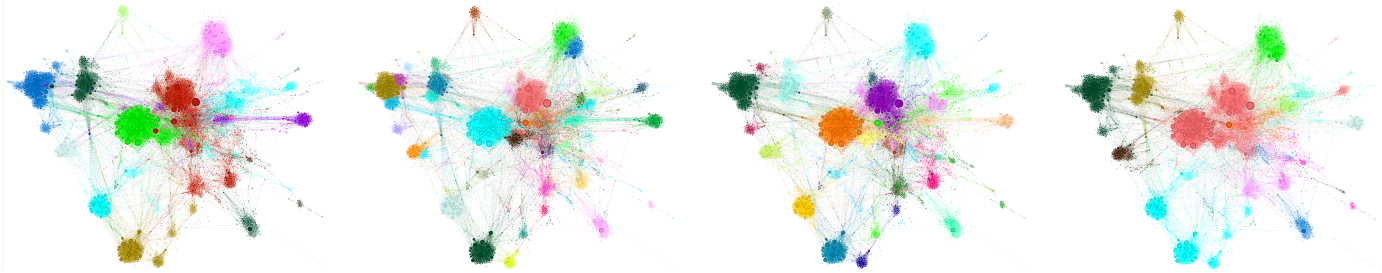
	Politicians Network						Google+ Network					
Ranking/Centrality	**DC**	**BC**	**CC**	**PC**	**EC**	**PR**	**DC**	**BC**	**CC**	**PC**	**EC**	**PR**
First node	1864	5800	5800	5800	5416	3008	136,198	-	-	-	116,002	157,237
Second node	4874	1864	4081	1864	1595	5800	5381	-	-	-	136,198	5381
Third node	5800	3576	2059	3576	4602	1864	116,002	-	-	-	159,894	173,914
Fourth node	5416	2900	4032	2900	4972	4874	145,647	-	-	-	130,830	125,121
Fifth node	1595	1324	3387	1324	1414	3576	66,836	-	-	-	35,778	116,648

5.3. Analysis Based on Community Detection

During the community detection experiments, our attention is also focused on the complexity of the algorithms, since in a real world scenario where a network could have millions of edges, any algorithm quadratic in n (especially, in l) could be impractical to use. According to the Girvan–Newman and Lancichinetti–Fortunato–Radicchi benchmarks [40],

the Infomap and Louvain algorithms have the best performance regarding running time and result accuracy. The Louvain and LPA algorithms are also built-in algorithms in the Neo4j graph database engine—being the first production ready algorithms of the engine (https://neo4j.com/docs/graph-algorithms/current/algorithms/community/, accessed on 12 July 2021).

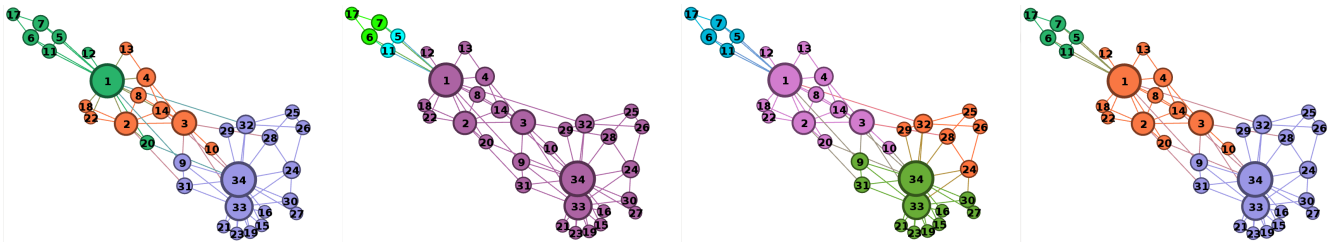
The results of different community detection algorithms on the Politician dataset are shown in Figure 1. It is interesting to analyze the subtle differences between these communities. The colors used for plot correspond to the community ID. It is important that visible structural differences can be observed, even on this moderately small network. CNM seems to yield a good community structure, however, the central red region is dispersed, which could be split into three different communities (this may be explained by the nature of the algorithm to merge two clusters—adjusting the parameter of the dendrogram cut will influence the final community structure). Furthermore, there are some outlier nodes, e.g., the red node in the leftmost blue region. The LPA solves the separation of the aforementioned central red community, but, there are cases where such separation may not be justified, e.g., the leftmost region, or the top right green and blue communities—again, this may be fine-tuned by varying the voting algorithm. Infomap yields the smallest number of communities, which can be seen on the plot, too, having many separate groups with the same color, i.e., community ID—the explanation may lie in the fact that the for small communities, the length of the Huffman coded path will not be any smaller by splitting these groups (the size of the communities together does not result in longer Huffman codes per node), on the contrary, it may become longer since a new entrance and exit code must be added to the path. Finally, Louvain communities seem to capture the network's community structure in the best possible way, with clear separations, but not in a excessive manner, as in the many cases of LPA.



(**a**) CNM (**b**) LPA (**c**) Louvain (**d**) Infomap

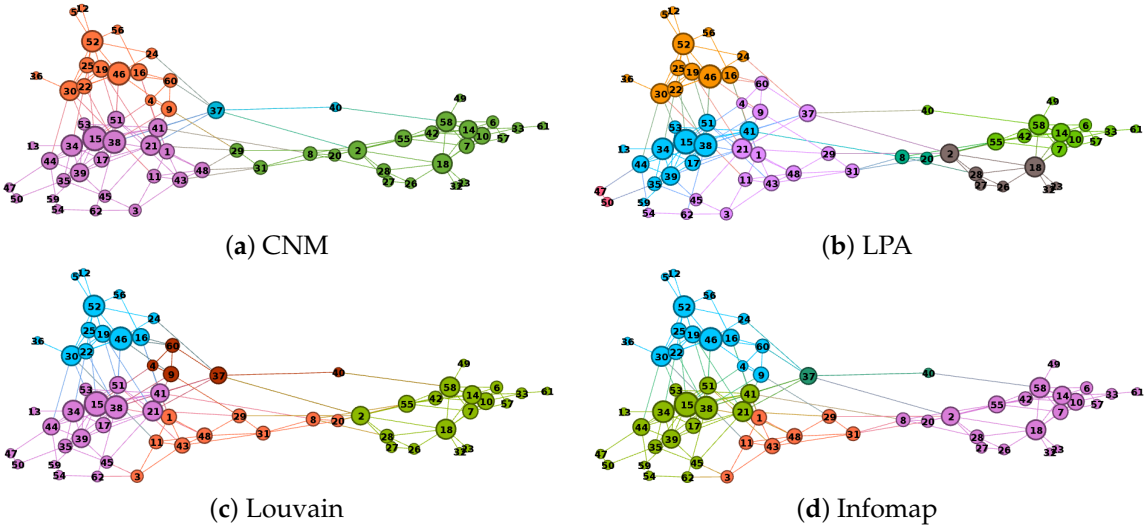
Figure 1. Comparison between different community detection algorithms on the Politicians network. Layout: Gephi's ForceAtlas 2.

These observations are valid regarding the analysis of the other networks as well, shown in Figures 2 and 3.



(**a**) CNM (**b**) LPA (**c**) Louvain (**d**) Infomap

Figure 2. Comparison between different community detection algorithms on the Karate network.



(**a**) CNM (**b**) LPA

(**c**) Louvain (**d**) Infomap

Figure 3. Comparison between different community detection algorithms on the Dolphin network.

The numerical statistics regarding the number of communities, their sizes, densities and topological properties are shown in Figure 4. Here we show several bar plots and line series. Each bar plot refers to the different community detection algorithm applied on the network: from left to right, respectively, CNM, LPA, Louvain, Infomap, and Infomap with a custom initial partition, that of the output of LPA. The algorithm's theoretic complexity is reflected in the first plot: CNM is by far the most time consuming, while LPA, Louvain and Infomap are the best in this regard. Also, the number of communities is way larger in case of the LPA compared to others—this can also be observed in their plots from Figure 1. Regarding modularity, coverage and performance scores, we can state that the custom partition-based Infomap does not outperform the simple one—quite on the contrary. The second row of plots show two line plots, each having five series according to the different algorithms, and two other plots to represent the community size histogram of Louvain. The first line series show max., average, median and min. size of the communities, while the second plot shows the same stats referring to their densities. We can observe that LPA produces the most dense communities, which can be justified by the larger number of communities, a more thorough separation of the nodes. These values originate from one evaluation—although these algorithms are non-deterministic, except the CNM, results across different runs showed insignificant, slightly observable differences.

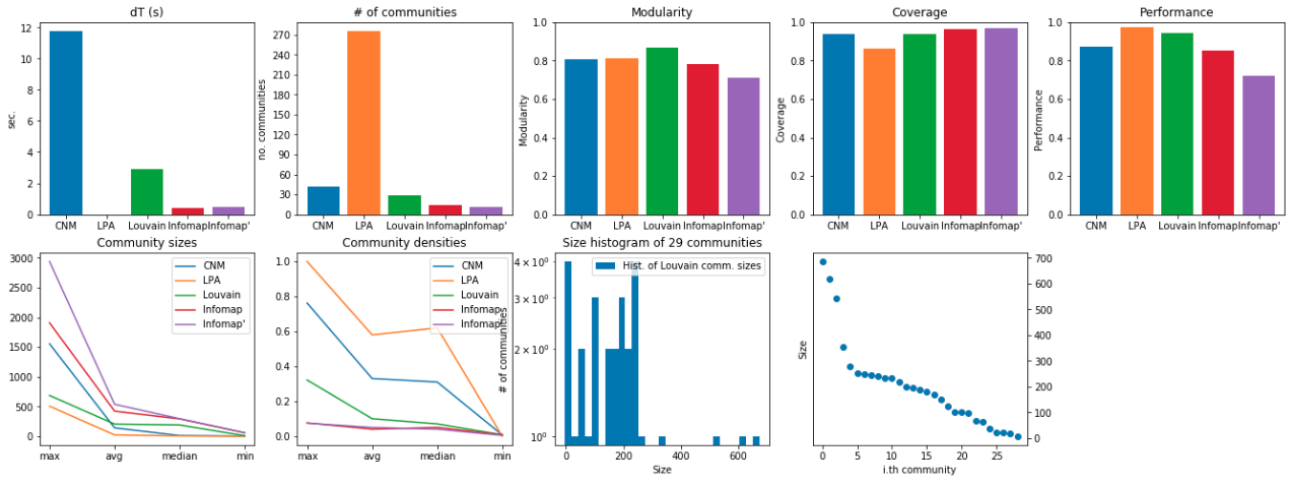


Figure 4. Community analysis of the Politicians network: the bar plots represent the runtime, number of communities, scores of modularity, coverage and performance, sizes of communities and their densities according to each algorithm. Additionally, a size histogram of Louvain communities is shown.

We list the hub-dominant nodes of each community according to each algorithm in Table 5 for the smallest networks and in Table 6 for the Politicians network.

Table 5. Hub dominant nodes of each community (node: HD score).

Karate Network				Dolphins Network			
CNM	**LPA**	**Louvain**	**Infomap**	**CNM**	**LPA**	**Louvain**	**Infomap**
n. 34: 0.875	n. 7: 1.0	n. 1: 1.0	n. 1: 0.9	n. 40: 1.00	n. 50: 1.00	n. 48: 0.83	n. 40: 1.00
n. 1: 0.85	n. 11: 1.0	n. 34: 1.0	n. 34: 0.875	n. 52: 0.64	n. 8: 1.00	n. 52: 0.82	n. 48: 0.83
n. 2: 0.75	n. 34: 0.6	n. 7: 0.75	n. 7: 0.75	n. 15: 0.45	n. 18: 0.83	n. 60: 0.75	n. 52: 0.64
		n 32: 0.6		n. 18: 0.43	n. 15: 0.82	n. 15: 0.53	n. 15: 0.53
					n. 52: 0.82	n. 18: 0.47	n. 18: 0.47
					n. 58: 0.73		
					n. 48: 0.43		

Table 6. Top hub-dominant nodes of the communities of Politician network (node: HD score).

CNM	**LPA**	**Louvain**	**Infomap**
node 4722: 1.00	node 990: 1.00	node 2386: 0.82	node 3008: 0.49
node 4552: 1.00	node 964: 1.00	node 3008: 0.79	node 5855: 0.38
node 3275: 1.00	node 902: 1.00	node 5699: 0.74	node 1965: 0.36
node 3254: 1.00	node 844: 1.00	node 5538: 0.72	node 4735: 0.33
node 3214: 1.00	node 770: 1.00	node 2469: 0.68	node 1140: 0.33
node 3115: 1.00	node 718: 1.00	node 3261: 0.65	node 5800: 0.30
node 2954: 1.00	node 669: 1.00	node 96: 0.64	node 2386: 0.28
node 2709: 1.00	node 5902: 1.00	node 5416: 0.62	node 4410: 0.25
node 212: 1.00	node 5856: 1.00	node 3173: 0.61	node 3950: 0.25
node 1688: 1.00	node 585: 1.00	node 1144: 0.59	node 112: 0.23
node 146: 1.00	node 5771: 1.00	node 191: 0.56	node 558: 0.21
node 1335: 1.00	node 5765: 1.00	node 98: 0.51	node 2698: 0.21
node 1130: 1.00	node 5649: 1.00	node 1965: 0.46	node 1864: 0.16
node 5363: 0.89	node 5648: 1.00	node 1864: 0.44	node 3048: 0.14

We also analyzed the distribution of node embeddedness and hub dominance. Figure 5 shows a scatter plot of the community sizes, scaled by their hub dominance score. It can be observed that the smaller the community, the larger the hub dominance, naturally. Cases where a large community has a large hub dominance are interesting scenarios, probably meaning the existence of important nodes inside the community. Similarly, the communities' node embeddedness histogram can be seen on the rightmost plot, demonstrating that this network's community structure results in highly embedded nodes, their degrees consisting of mostly inner connections of the community.

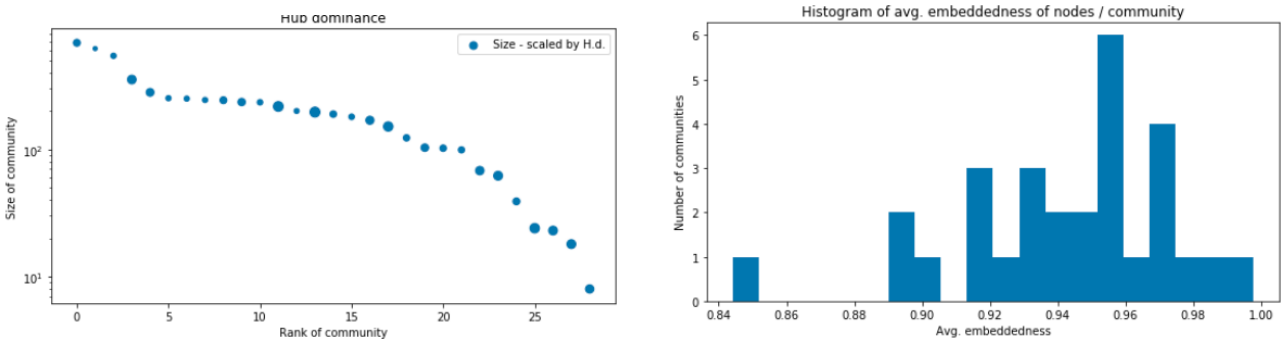
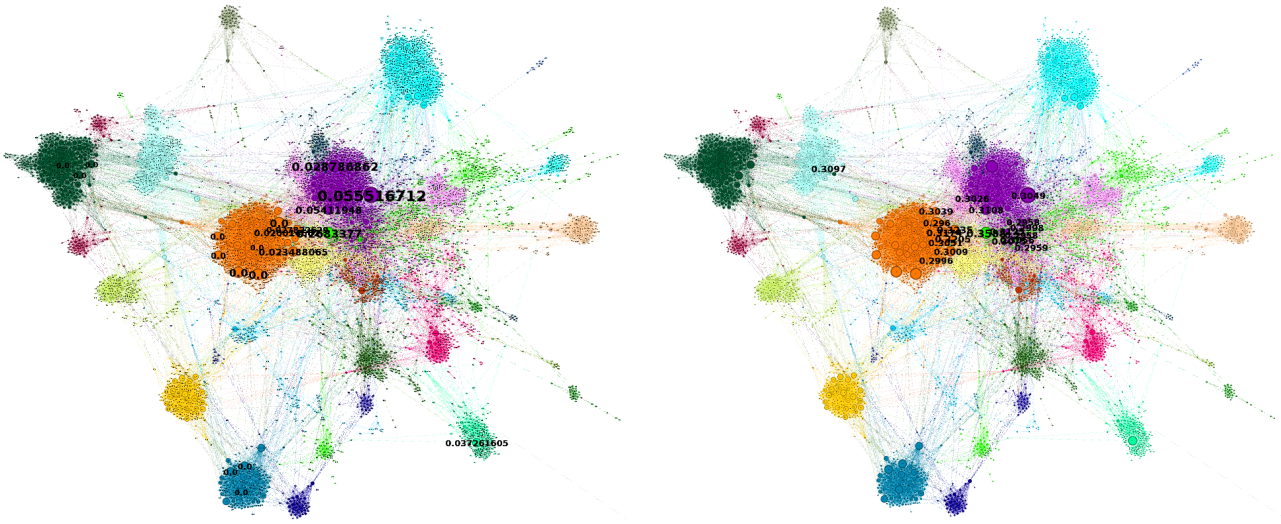


Figure 5. Politicians network: community sizes scaled by their hub dominance score, and histogram of node embeddedness.

5.4. Fusion of Node Centralities and Louvain Communities

We conclude our experiments by a fusion of the results yielded by Louvain (due to its ideal modularity structure, as depicted in Figure 1), and those obtained by calculating different node-level centrality metrics. Figure 6 shows Louvain communities of the Politicians network, nodes scaled according to their degree, and for each centrality metric, the top 20 nodes are extracted, their scores being scaled by the value of the respective metric. By visual analysis, one can observe that a series of centrality measures yield overlapping node sets. For example, the central orange group of nodes, with high degree nodes on its periphery, is detected by betweenness, closeness, page rank, eigen centralities as top 20 from the entire network. Furthermore, one of them is exactly the hub in the respective community—having hub dominance score of 0.6222. Another example refers to the central green node, being the hub-dominant node of that green community, which on the other hand is detected by all other centrality metrics, having outstanding closeness and page rank scores. These observations are formulated numerically as well, in Table 7. In this table, we list those hub-dominant nodes of the respective network, which appear in the top 20 centralities—listing their rank in parenthesis (i.e., 1 meaning that the node has the highest centrality score in the network). Table 7 indicates that the hub-dominant nodes in the Karate network are nodes 1, 6, 34 and 25—nodes 1 and 34 having top centrality metrics as well (see Section 5.2). Their hub-dominance score is also among the highest of these four hub nodes. It is interesting to notice that although node 25 is a hub node, its dominance score is only 0.6 (compared to 0.9 of node 1), and not any of the centrality measures ranks this node as top important one—similarly to the case of node 6.



(**a**) Betweenness centrality (**b**) Closeness centrality

Figure 6. Politicians: Comparison of centrality metrics and hub-dominance of communities.

Table 7. Karate network: Centrality scores of hub-dominant nodes—rank in paranthesis.

Node	DC	BC	CC	PC	EC	PR	K-Core	HD
1	16 (2)	0.4376 (1)	0.569 (1)	0.4376 (1)	0.3555 (2)	0.097 (2)	1	0.9091 (2)
6	4 (11)	0.03 (10)	0.3837 (17)	0.03 (10)	–	0.0291 (12)	–	0.75 (3)
34	17 (1)	0.3041 (2)	0.55 (3)	0.3041 (2)	0.3734 (1)	0.1009 (1)	1	1.00 (1)
25	–	0.0022 (18)	–	0.0022 (18)	–	0.0211 (19)	–	0.6 (4)

Furthermore, the susceptible-infected-recovered (SIR) model [41] could be engaged to investigate the spreading dynamics in the network considering the important nodes determined by the previous centrality and community based approaches. The propagation ability of a node could be representative to be studied in this context. In Figure 7 we

compare the evolution in time of the information spreading for the node having highest betweenness centrality score (left) and first node for hub dominance (right). The green nodes are the ones not containing any information (susceptible), red containing information (infected) and grey are the nodes that had information, but managed to recover from it (recovered). The graphs attached to each frame are showing in time the evolution of each type of node (green, red and gray). Figure 7 only depicts the first iterations to illustrate how the information spreads from the initial infected nodes to the other nodes. These empirical results have been obtained for a particular SIR model with the probability of 0.2 for an infected node to infect its neighbors and a recovery rate of 1.2. We can note that, if we compare every frame between the two nodes, they are both confirmed as important nodes in this small network; however, a deep statistical validation is still required in order to assess the importance of high ranked centrality nodes and hub dominant ones, further taking into account the stochastic nature of the spreading models.

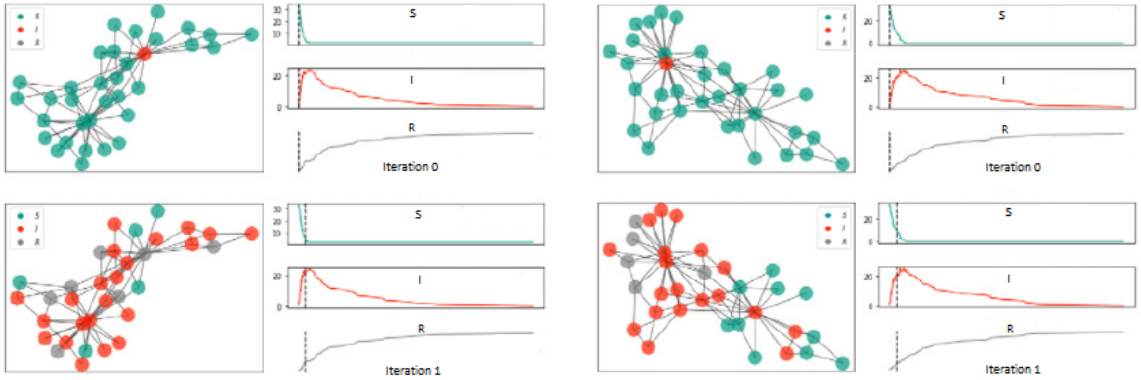


Figure 7. Karate network: using the SIR model for node 1 (first betweenness centrality node—**left**) and node 34 (first hub dominance node—**right**).

5.5. Benchmark Testing on Lancichinetti–Fortunato–Radicchi and Erdős–Rényi Networks

In order to validate our hypothesis of the correlation between hub-dominant nodes and important nodes having top centrality metrics, we carried out experiments on different sized LFR networks [38] and ER networks [39]—following the logical flow of the above described analysis, but now on much more randomly generated networks. The steps are as follows: rank the nodes according to centrality metrics, determine the community structure of the network with Louvain algorithm, then determine the hub-dominant nodes of each community and finally, verify if these hub nodes are among top centrality nodes according to any of the mentioned centrality criteria. We mention that the LFR networks already have the planted partition structure. In order to show consistent results across the two random network generation models, regardless of the underlying partitioning method, we use the Louvain partitioning of the LFR networks (instead of their planted partitions), just as in the case of ER networks.

The importance of using both of these network models is to address RQ3. Since ER networks do not have an intrinsic community structure—as stated by the random hypothesis, also described in [2], the joint results of the experiments run on LFR and ER models do confirm that hub-dominant nodes tend to be among top important nodes, especially on lightweight networks. The importance of the random hypothesis in this context is that regardless of the networks's nature of having a inherent partition or not, running the Louvain algorithm, the obtained hub nodes will be among those having highest centrality metrics.

The LFR networks were generated by using the following parameters: degree distribution's power-law exponent 3, community-size distribution's power-law exponent 1.1, fraction of inter-community edges incident to each node 0.1. The degree intervals and minimum number of communities (c_{min}) are shown in Table 8. The ER networks were

generated using the $G(n, p)$ random model, with n of 10, 50 and 100, and p being 0.3 and 0.8.

Table 8. Parameters of LFR network generation.

Nodes	c_{min}—Min. Number of Communities	d_{min}—Min. Degree	d_{max}—Max. Degree
10	2	2	5
50	5	5	10
100	10	10	30
500	10	10	30

Tables 9 and 10 show the results of this experiment, according to four different sized network categories. Each of these shows the percentage of hub-dominant nodes being among top k nodes according to the respective centrality criteria. The process of finding this percentage is described in the following. For each of these four network categories, several instances were generated—each instance having a set of hub-dominant nodes, and the ranking of important nodes. The respective percentage is calculated by taking into account all of these network instances, and determining how many of these instances' hub nodes were among the corresponding top ranking—separately, for top 1–3, top 4–6 and top 7–10 nodes. Due to the computational complexity of some node centrality metrics (e.g., betweenness centrality), for larger networks only a small number of instances could be evaluated—i.e., only 7 networks of size $n = 500$. Thus, results corresponding to larger networks may be biased, but we can still formulate our observation, based on the comparison with smaller networks.

Although some of the experiments were run on statistically too few networks in the case of LFR model, the Erdős–Rényi experiment was carried out on a statistically representative sample size—1000 networks for each parametrizations. We may conclude that for small networks, the hub-dominant nodes are among the top important nodes of the network, but as the network grows, these hubs do not necessarily have top centrality metrics—hence, a comprehensive analysis of the network should not be limited to the different node-level centrality metrics, but shall keep in mind the network's community partitioning as well.

Table 9. Average percentage of HD nodes being top important nodes in different sized LFR networks.

	n = 10 (91 Networks)			n = 50 (50 Networks)		
Metric	**Top 1–3**	**Top 4–6**	**Top 7–10**	**Top 1–3**	**Top 4–6**	**Top 7–10**
BC	59%	25.4%	5.9%	10.9%	7.7%	10.1%
CC	56.9%	15.6%	14.1%	13.2%	7.4%	7.9%
DC	69.8%	7.1%	1.6%	22.3%	8.4%	7.0%
PC	59.3%	26.2%	5.9%	10.9%	7.7%	10.1%
EC	48.7%	15.3%	28.6%	16.7%	2.3%	3.7%
PR	76.9%	13.1%	2.6%	20.8%	10.6%	13.4%
	n = 100 (30 Networks)			**n = 500 (7 Networks)**		
BC	8.9%	6.2%	8.4%	2.2%	1.1%	2.5%
CC	12.8%	7.2%	6.7%	2.2%	0.5%	1.6%
DC	20.1%	5.9%	7.8%	4.4%	1.0%	1.7%
PC	8.9%	6.2%	8.4%	2.2%	1.1%	2.5%
EC	18.8%	0%	1.4%	2.7%	1.0%	0.6%
PR	19.2%	8.3%	8.4%	4.7%	2.1%	2.7%

Table 10. Average percentage of HD nodes being top important nodes in different sized Erdős–Rényi networks of model $G(n, p)$.

					$G(n, 0.3)$ **Model**				
	n = 10 (1000 Networks)			**n = 50 (1000 Networks)**			**n = 100 (1000 Networks)**		
Metric	**Top 1–3**	**Top 4–6**	**Top 7–10**	**Top 1–3**	**Top 4–6**	**Top 7–10**	**Top 1–3**	**Top 4–6**	**Top 7–10**
BC	47.6%	25.2%	13.6%	18.4%	13.1%	13.0%	12.4%	8.9%	9.2%
CC	41.0%	22.3%	18.2%	20.7%	12.1%	11.3%	14.3%	8.7%	8.2%
DC	45.5%	17.5%	8.9%	20.9%	11.8%	11.7%	14.4%	8.7%	8.2%
PC	47.4%	26.0%	13.7%	18.4%	13.1%	13.0%	12.4%	8.9%	9.2%
EC	38.2%	27.1%	24.1%	17.4%	13.4%	13.2%	12.5%	8.7%	9.2%
PR	49.1%	25.3%	15.1%	19.3%	14.0%	13.6%	13.3%	9.4%	9.2%
					$G(n, 0.8)$ **Model**				
BC	44.2%	25.5%	27.1%	13.7%	10.5%	12.3%	10.5%	7.5%	7.8%
CC	52.5%	18.6%	18.0%	17.3%	11.4%	10.5%	12.8%	7.4%	8.0%
DC	52.5%	18.6%	18.0%	17.3%	11.4%	10.5%	12.8%	7.4%	8.0%
PC	44.2%	25.5%	27.0%	13.7%	10.5%	12.3%	10.5%	7.5%	7.8%
EC	44.5%	25.0%	28.5%	13.0%	10.9%	12.8%	10.8%	7.2%	8.0%
PR	44.4%	25.9%	28.2%	13.8%	10.6%	12.5%	10.9%	7.2%	8.2%

5.6. Final Remarks

According to the above discussed results, we can formulate the following responses to the aforementioned research questions. The different node centrality metrics select different nodes to be important, no predefined method can be selected as the rightmost criteria and the decision must be made based on the application context.

Community detection helps the analysis in several ways. As some of the prevalent algorithms' complexity is close to linearity, this kind of analysis may be more efficient than determining node importance in the global context of the network. Furthermore, both the combined plots of the communities and centralities, the tabular data about the fusion of hub-dominant nodes and top centrality scores and the benchmark tests as well suggest that there is a relation between the community-based important nodes (i.e., hub-dominant nodes) and several centrality criteria; however, only in lightweight networks. The observation can be motivated with the fact that as a network grows, a simple node partitioning may not have full vision on the structure of the network, thus, node centrality metrics based on full-graph traversals and paths yield other important nodes than by selecting just the representative items of each community.

6. Conclusions and Future Work

This paper analyzes different methods for node importance ranking using centrality scores and community detection, then investigates the relation of these methods by carrying out multiple experiments on both real and synthetic networks. In a complex network, each node centrality metric offers a different perspective on the dataset and the most precise analysis can be obtained by taking into consideration not only a multitude of these metrics, but also the community structure of the whole network. In order to verify this claim, several experiments have been carried out on four real world networks and different sized random networks, and the top important nodes have been examined. The results from centrality measures and community detection show that these methods yield common important nodes and further suggest that the relation between hub-dominant nodes determined through community detection and highly-ranked centrality nodes is valid particularly for lightweight networks.

The proposed approach is relevant to undirected and unweighted networks as well. Since the experimental results confirm that there are some important nodes that are found by both measurements, future studies are needed to focus on an aggregate approach of dynamic characteristics and network structure in order to find the node significance. By

determining node chaining patterns, one could obtain interesting insights to the nature of the network. These patterns may be the subject of further network motif analysis or graph convolutional neural network application.

Author Contributions: Conceptualization, C.C., L.D., B.-E.-M.M., A.P. and A.M.; methodology, A.P. and A.M.; software, A.P. and A.M.; validation, C.C., L.D. and B.-E.-M.M.; writing—original draft preparation, A.P. and A.M.; writing—review and editing, C.C. and L.D.; visualization, A.P. and A.M.; supervision, C.C., L.D. and H.G.; project administration, C.C.; funding acquisition, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by a grant of the Romanian Ministry of Education and Research, CCCDI–UEFISCDI, project number PN-III-P2-2.1-PED-2019-2607, within PNCDI III.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This work was supported by a grant of the Romanian Ministry of Education and Research, CCCDI–UEFISCDI, project number PN-III-P2-2.1-PED-2019-2607, within PNCDI III.

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

References

1. Estrada, E. *The Structure of Complex Networks: Theory and Applications*; Oxford Scholarship Online; Oxford University Press: Oxford, UK, 2011.
2. Barabási, A.L.; Pósfai, M. *Network Science*; Cambridge University Press: Cambridge, UK, 2016.
3. Newman, M. *Networks*, 2nd ed.; Oxford University Press: Oxford, UK, 2018.
4. Pizzuti, C.; Socievole, A. Computation in Complex Networks. *Entropy* **2021**, *23*, 192. [CrossRef]
5. Omar, Y.M.; Plapper, P. A Survey of Information Entropy Metrics for Complex Networks. *Entropy* **2020**, *22*, 1417. [CrossRef]
6. Li, X.; Sun, Q. Identifying and Ranking Influential Nodes in Complex Networks Based on Dynamic Node Strength. *Algorithms* **2021**, *14*, 82. [CrossRef]
7. Ullah, A.; Wang, B.; Sheng, J.; Long, J.; Khan, N.; Sun, Z. Identification of nodes influence based on global structure model in complex networks. *Sci. Rep.* **2021**, *11*. [CrossRef]
8. Zhu, J.; Wang, L. Identifying Influential Nodes in Complex Networks Based on Node Itself and Neighbor Layer Information. *Symmetry* **2021**, *13*, 1570. [CrossRef]
9. Wasserman, S.; Faust, K. Social network analysis: Methods and applications. In *Structural Analysis in the Social Sciences*; Cambridge University Press: Cambridge, UK, 1994; p. 868.
10. Lancichinetti, A.; Fortunato, S. Community detection algorithms: A comparative analysis. *Phys. Rev. E* **2009**, *80*, 056117. [CrossRef]
11. Opsahl, T.; Agneessens, F.; Skvoretz, J. Node centrality in weighted networks: Generalizing degree and shortest paths. *Soc. Netw.* **2010**, *32*, 245–251. [CrossRef]
12. Bauer, F.; Lizier, J.T. Identifying influential spreaders and efficiently estimating infection numbers in epidemic models: A walk counting approach. *EPL Europhys. Lett.* **2012**, *99*, 68007. [CrossRef]
13. Hansen, D.; Shneiderman, B.; Smith, M.; Himelboim, I. Twitter: Information flows, influencers, and organic communities. In *Analyzing Social Media Networks with NodeXL*; Morgan Kaufmann: San Mateo, CA, USA, 2020; pp. 161–178.
14. Golbeck, J. Network structure and measures. In *Analyzing the Social Web*; Elsevier: Amsterdam, The Netherlands, 2013.
15. Langville, A.N.; Meyer, C.D. *Google's PageRank and Beyond: The Science of Search Engine Rankings*; Princeton University Press: Princeton, NJ, USA, 2006.
16. Miorandi, D.; Pellegrini, F.D. K-Shell decomposition for dynamic complex networks. In Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, Avignon, France, 31 May–4 June 2010.
17. Fortunato, S. Community detection in graphs. *Phys. Rep.* **2009**, *486*, 75–174. [CrossRef]
18. El-Moussaoui, M.; Agouti, T.; Tikniouine, A.; Adnani, M.E. A comprehensive literature review on community detection: Approaches and applications. *Procedia Comput. Sci.* **2019**, *151*, 295–302. [CrossRef]
19. Schaeffer, S. Graph clustering. *Comput. Sci. Rev.* **2007**, *1*, 27–64. [CrossRef]
20. Malliaros, F.; Vazirgiannis, M. Clustering and community detection in directed networks: A survey. *Phys. Rep.* **2013**, *533*, 95–142. [CrossRef]
21. Ahajjam, S.; Mohamed, E.H.; Hassan, B. A new scalable leader-community detection approach for community detection in social networks. *Soc. Netw.* **2018**, *54*, 41–49. [CrossRef]
22. Ravasz, E. Hierarchical organization of modularity in metabolic networks. *Science* **2002**, *297*, 1551–1555. [CrossRef] [PubMed]
23. Girvan, M.; Newman, M. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [CrossRef]

24. Pons, P.; Latapy, M. Computing communities in large networks using random walks. *J. Graph Algorithms Appl.* **2006**, *10*, 191–218. [CrossRef]
25. Rosvall, M.; Axelsson, D.; Bergstrom, C. The map equation. *Eur. Phys. J. Spec. Top.* **2009**, *178*, 13–23. [CrossRef]
26. Clauset, A.; Newman, M.; Moore, C. Finding community structure in very large networks. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **2005**, *70*, 066111. [CrossRef]
27. Blondel, V.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
28. Guimera, R.; Amaral, L.A.N. Functional cartography of complex metabolic networks. *Nature* **2005**, *433*, 895–900. [CrossRef]
29. Boettcher, S.; Percus, A.G. Optimization with extremal dynamics. *Phys. Rev. Lett.* **2001**, *86*, 5211–5214. [CrossRef] [PubMed]
30. Duch, J.; Arenas, A. Community detection in complex networks using extremal optimization. *Phys. Rev. E* **2005**, *72*, 027104. [CrossRef]
31. Raghavan, N.; Albert, R.; Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **2007**, *76*, 036106. [CrossRef] [PubMed]
32. Shahrivari Joghan, H.; Bagheri, A. Local edge betweenness based label propagation for community detection in complex networks. In Proceedings of the 2017 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 14–16 December 2017.
33. Shahrivari Joghan, H.; Bagheri, A.; Azad, M. Weighted label propagation based on local edge betweenness. *J. Supercomput.* **2019**, *75*, 8094–8114. [CrossRef]
34. Brandes, U. A faster algorithm for betweenness centrality. *J. Math. Sociol.* **2004**, *25*, 163–177. [CrossRef]
35. Newman, M.E.; Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **2004**, *69*, 026113. [CrossRef] [PubMed]
36. Brandes, U.; Delling, D.; Gaertler, M.; Gorke, R.; Hoefer, M.; Nikoloski, Z.; Wagner, D. On modularity clustering. *IEEE Trans. Knowl. Data Eng.* **2008**, *20*, 172–188. [CrossRef]
37. Orman, G.; Labatut, V.; Cherifi, H. Comparative evaluation of community detection algorithms: A topological approach. *J. Stat. Mech. Theory Exp.* **2012**, *2012*, P08001. [CrossRef]
38. Lancichinetti, A.; Fortunato, S.; Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **2008**, *78*, 046110. [CrossRef]
39. Erdös, P.; Rényi, A. On Random Graphs I. *Publ. Math. Debr.* **1959**, *6*, 290.
40. Yang, Z.; Algesheimer, R.; Tessone, C. A comparative analysis of community detection algorithms on artificial networks. *Sci. Rep.* **2016**, *6*, 30750. [CrossRef] [PubMed]
41. Kakehashi, M.; Kawano, S. Fundamentals of mathematical models of infectious diseases and their application to data analyses. In *Handbook of Statistics*; Elsevier: Amsterdam, The Netherlands, 2017; Volume 36, pp. 3–47.