Understanding Vulnerability of Communities in Complex Networks

Arindam Pal
The Commonwealth Scientific and Industrial Research Organisation
56 PUBLICATIONS **324** CITATIONS

Ponnurangam Kumaraguru
International Institute of Information Technology, Hyderabad
261 PUBLICATIONS **5,805** CITATIONS

Tanmoy Chakraborty
Indraprastha Institute of Information Technology
238 PUBLICATIONS **1,833** CITATIONS

On the Vulnerability of Community Structure in Complex Networks

Viraj Parimi[1*], Arindam Pal[3,4,6], Sushmita Ruj[3,5], Ponnurangam Kumaraguru[2], and Tanmoy Chakraborty[2]

[1] Carnegie Mellon University, Pittsburgh, USA
[2] IIIT-Delhi, New Delhi, India
[3] Data61, CSIRO, Sydney, Australia
[4] Cyber Security CRC, Canberra, Australia
[5] Indian Statistical Institute, Kolkata, India
[6] University of New South Wales, Sydney, Australia

Abstract. In this paper, we study the role of nodes and edges in a complex network in dictating the robustness of a community structure towards structural perturbations. Specifically, we attempt to identify all vital nodes, which, when removed, would lead to a large change in the underlying community structure of the network. This problem is critical because the community structure of a network allows us to explore deep underlying insights into how the function and topology of the network affect each other. Moreover, it even provides a way to condense large networks into smaller modules where each community acts as a meta node and aids in more straightforward network analysis. If the community structure were to be compromised by either accidental or intentional perturbations to the network, that would make such analysis difficult. Since identifying such vital nodes is computationally intractable, we propose a suite of heuristics that allow to find solutions close to the optimality. To show the effectiveness of our approach, we first test these heuristics on small networks and then move to more extensive networks to show that we achieve similar results. Further analysis reveals that the proposed approaches are useful to analyze the vulnerability of communities in networks irrespective of their size and scale. Additionally, we show the performance through an extrinsic evaluation framework – we employ two tasks, i.e., link prediction and information diffusion, and show that the effect of our algorithms on these tasks is higher than the other baselines.

Keywords: Community Structure, Vulnerability Assessment, Complex Networks

1 Introduction

A large body of research in complex networks involves the study and effects of community structure as it is one of the salient structural characteristics of real-world networks. A network is said to have a community structure if it can be

[*] The work was done when Viraj was an undergraduate student at IIIT-Delhi, India.

grouped easily into sets of nodes. Each set of nodes is densely connected internally and sparsely linked externally. Research in this field is broadly classified into two categories – first, where one detects the community structure within a given network and the other where one studies the properties of a community structure to infer more details about the network. A variety of methods have been proposed that target the former issue as described [38,15]. The advantage of such algorithms is that it provides us with an efficient and approximate clustering of nodes that allows us to condense large networks to smaller ones owing to their mesoscopic structure. Within the second paradigm, the ability to detect vital nodes is of significant practical importance. It provides insight into how a network functions and how the network topology change affects the interactions between the nodes within the network. Exploring this structural vulnerability of the network allows us to prepare beforehand if the network is affected by undesired perturbations and adversarial attacks. A significant factor in understanding this is to analyze the network and comprehend the effect of these vital nodes' failure on the community structure of the network.

In this paper, we attempt to identify and investigate some vital nodes in a network, whose removal highly affects the network's community structure. Formally, given a network $G(V, E)$ and a positive integer k, we intend to find a set $S \in V$ consisting of k nodes whose removal leads to the maximum damage of the community structure. The change in the community structure is quantified using different measures such as Modularity [45], Normalized Mutual Information [24], Adjusted Rand Index [36], etc.

There are many real-world applications of this problem. Consider a power grid network where a power outage is a frequently occurring event. Most power networks have a regional hub that caters to the needs of nearby power stations. In such a scenario, the vendor of this power grid needs to make quick decisions about how the failure of some nodes in the network would affect the customers. The solution would be to ensure that crucial nodes in this network have enough backup so that the restoration process can move effortlessly. Another application would be the railway networks, where inadvertent cutting of routes to certain stations can cause significant problems for the city residents. Hence, the government needs to ensure that routes to certain critical stations have redundancies so that if one route gets cut off, then the trains can utilize other routes. This problem also has applications in Online Social Networks such as the worm containment problem [42]. This knowledge would provide helpful insights into protecting sensitive nodes once worms spread out into the network. In all of the issues mentioned above, it is evident that one needs to study the structural integrity of the communities underlying in the network. Note that a minor structural change that can be as small as removing a node in the network can lead to the community's breakdown that the node was a part of, given that the removed node had a considerable influence on the network. If the removed node were of less significance, that would have less impact on the network's community structure.

Additionally, understanding the network vulnerability from the standpoint of the community structure is essential in real-world settings. The networks that are dealt with here have tremendous size, which adds to the computational overhead and, most importantly, shed light on some latent characteristics shared by different nodes. Since communities can act as meta-nodes, they allow for a more comfortable study of large networks. This reduces the computational overhead and provides useful insights based on the properties shared by the community's nodes that can be exploited to understand the network's vulnerability.

We propose a hierarchical greedy approach that selects communities based on the community-centric properties in phase 1 and then, within that community, selects the most vulnerable nodes in phase 2. We test this algorithm on six real-world datasets of varying sizes. Our empirical results indicate that the algorithm can identify properties that contribute most towards community structures' vulnerabilities in a network. The past work in this domain [64,3] is restricted to smaller networks, but our work extends the scope towards even more extensive networks with the number of nodes in the order of millions.

In summary, our contributions in this paper are as follows:

- We study the structural vulnerability of communities in networks and assess the impact of nodes' failure on the underlying community structure.
- We suggest few heuristics, including a hierarchical greedy approach that allows for identifying such critical nodes in the network that profoundly impact the community structure.
- We conduct experiments on real-world datasets and show the effectiveness of the heuristics that we propose.
- We propose a novel task-based strategy to evaluate the extent of correctness of the algorithm extrinsically. This allows us to estimate the performance of our algorithm in a real-world context.

The remaining part of this paper is as follows. We discuss the literature review on community detection and vulnerability assessment in Section 2. We formalize our problem in Section 3. We then discuss some preliminaries in Section 4. We present our proposed methodology in Section 5. Section 6 describes the datasets used to evaluate the proposed approach. In Section 7 we provide the results of our method when applied to these datasets and briefly discuss the evaluation strategy how we go about validating our proposed method on larger datasets. We put forward our conclusion in Section 8.

2 Related Work

This section first presents the literature on community detection algorithms and then discusses community vulnerability analysis.

2.1 Community Detection

Community detection, a task of grouping similar nodes together, is a significant problem. A lot of work has been done in the past to come up with a solution effectively. Numerous approaches have been developed and applied to detect community structure. For instance, a hierarchical agglomerative algorithm was proposed by Newman et al.[31]. An extensive literature survey can be found in [38]. Here we briefly mention some of the popular approaches.

Initial efforts at community detection assumed that the nodes are densely connected within a community and sparsely connected across communities. Under this assumption, the algorithms proposed were targeted towards community detection in static networks. Such efforts involved several approaches such as modularity optimization [10,23,33,45,48], clique percolation [27,52], information-theoretic approaches [59,60], and label propagation [53,65,66]. Furthermore, spectral partitioning [46,56], local expansion [7,39], random-walk based approaches [25,37], diffusion-based approaches [53] and significance-based approaches [40] were explored to help in identifying the community instances within a network. Several pre-processing methods [5,57] were also introduced to improve upon these algorithms. Such methods involved generating a preliminary community structure on a set of nodes and modifying iteratively until all the nodes are covered. Apart from these, several other algorithms were proposed to detect communities in dynamically evolving networks [2,58].

Another set of community detection algorithms allows a vertex to be part of multiple communities simultaneously. Such overlapping community detection algorithms used ideas based on local expansion and optimization. These include RankRemoval [8] which uses local density function, LFM [39], and MONC [34] which iteratively maximize a fitness function, and GCE [41] which makes use of an agglomerative pipeline to detect overlapping community instances. Other approaches also looked into the idea of partitioning links instead of nodes to discover the network's underlying community structure. The clique percolation method was also explored in CFinder [1], but since many real-world networks are sparse, these methods generally produced low-quality outputs. Recently, several new ideas were presented, such as [44] which solved a constrained optimization problem using simulated annealing techniques, and [47,51,71,55] which used mixture models to solve the problem. Even a game-theoretic approach [22] was proposed in which a community is equated to a Nash local equilibrium. Non-negative Matrix Factorization [68,72] framework has also been utilized to identify fuzzy or overlapping community structures. Chakraborty et al. proposed MaxPerm and GenPerm, two greedy approaches which maximize a node-centric metric, called ”permanence” to detect disjoint [31] and overlapping communities [17]. They also proposed a post-processing technique based on permanence to detect overlaps from a disjoint community structure [14]. Several ensemble-based approaches were also proposed by leveraging the output of disjoint community detection methods [18,16,19].

2.2 Community Vulnerability Analysis

Assessing the structural network vulnerability has received increasing attention. For example, Nguyen et al. [50] have proposed a Community Vulnerability Assessment (CVA) problem and suggested multiple heuristic-based algorithms based on the modularity measure of communities in the network. These approaches are restricted to the scope of online social networks and do not cater to general network structures. Another work by Nguyen et al. [49] explored the number of *connected triplets* in a network as they capture the strong connection of communities in social networks. They proposed an efficient approximation algorithm to identify triangle breaking points like nodes or links within a network.

Additionally, different measures and metrics have been proposed to measure the robustness of a network. Such efforts include the average size of a cluster, relative size of the largest components, diameter, and network connectivity. One approach dealt with this problem using the weighted count of loops in a network. Chan et al. [21] addressed this problem in both deterministic and probabilistic settings where they suggested solutions based on minimum node cutset. Frank et al. [29] outlined a solution that uses the second smallest eigenvalue of a Laplacian matrix of a network and termed it as the algebraic connectivity of that network. Fiedler [28] proposed four basic attack strategies, namely, ID removal, IB removal, RD removal, and RB removal. ID and RD removal deal with the degree distribution of the network. The only difference is that the second approach changes the removal strategy based on the degree distribution change. IB and RB removal are also similar constructs, but they are based on the betweenness distribution. Holme et al. [35] used an algorithm adapted from Google's PageRank providing a sequence of losses that add to the collapse of the network. Allesina et al. [4] evaluated the network characteristics like cyclomatic number and gamma index. They mentioned that such global graph-theoretic indices are not sufficient to measure a network's vulnerability, but they showcase the hierarchy of nodes in the system.

Ramirez et al. [54] proposed an approach where the community structure's resilience is quantified by introducing disruption in the original network and measuring the change in the community structure temporally, i.e., after the disconnection and during the restoration process. Geubesic et al. [32] provided a review of various approaches that use the *facility importance* concept to understand the system-wide vulnerability. These concepts include alpha index, beta index etc. They concluded that simple graph-theoretic measures were not sufficient to measure the vulnerability of a network. It also required many local efforts, such as the degree of node. They mentioned that global indicators measure network accessibility, path availability, and local measures to provide better information about node criticality. Sankaran et al. [64] proposed a new vulnerability metric where they considered a combination of external and internal factors such as connection density. They proposed a non-linear weighted function to combine these factors. However, the proposed method was not proved feasible in practice as the weights of all the elements were assumed to be equal and not self-adjusting to the network. These methods allow us to quantify a community's vulnerability but do not provide us with a set of nodes that contribute to the community's vulnerability.

The information of critical nodes that contribute to the communities' vulnerability would provide far more insights than just discovering the vulnerable community. As a result, a more comprehensive study is required to assess the vulnerability of general network structures.

3 Problem Statement

Let $G(V, E)$ be an input graph and let k be the number of nodes that we want to select. Let $\mathcal{A}$ be a community detection algorithm. For a vertex set $S \in V$, let $G[S]$ be the subnetwork induced by S and $f(\mathcal{A}(G[V]), \mathcal{A}(G[V \setminus S]))$ is a value function that computes some measure of the difference between the community structures of $G[V]$ and $G[V \setminus S]$ obtained from $\mathcal{A}$. We need to identify a set $S \in V$ of size k which,

$$\textbf{maximize} \quad f(\mathcal{A}(G[V]), \mathcal{A}(G[V \setminus S])) \tag{1}$$

This problem is computationally intractable as shown by Alim et al. [50] and hence requires the use of greedy heuristics to approach the optimal answer.

4 Preliminaries

We used the Louvain algorithm for detecting the underlying community structure. It is a greedy optimization algorithm proposed by Blondel et al. [10], that tries to optimize the modularity metric of a network and extracts communities from large networks using heuristics. This approach, however, can easily be modified to use with other community detection algorithms as well.

To quantify the difference between the community structures of $G[V]$ and $G[V \setminus S]$, we use the following measures:

• **Modularity:** It is a measure to quantify the strength of the division of the network into communities. Networks with high modularity have denser connections within a community and sparse connections across communities. It is defined as follows,

$$Q = \frac{1}{(2m)} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{(2m)} \right] \delta(c_v, c_w), \tag{2}$$

where m = number of edges, A = adjacency matrix, k_v = degree of node v, c_v = community label of node v, $\delta(c_v, c_w) = 1$ if $c_i = c_j$ and 0 otherwise.

• **Normalized Mutual Information:** It is a measure that quantifies the similarity between two community structures. It produces 1 if two community structures are exactly the same and 0 otherwise. It is defined as follows.

$$N = \frac{2 \sum_{i=1}^{c_X} \sum_{j=1}^{c_Y} \left[n_{ij} \log \left(\frac{n_{ij} n}{x_i y_j} \right) \right]}{(n-k) H(X) + \bar{y} \log(n-k) - \sum_{j=1}^{c_Y} (y_j \log(y_j))}, \tag{3}$$

where c_X = number of communities in community structure X, c_Y = number of communities in community structure Y, $n_{ij} = |X_i \cap Y_j|$, n = number of nodes in the network, $x_i = |X_i|$, $y_i = |Y_i|$, $\bar{y}$ = total size of communities in Y, $H(X)$ = entropy of X.

• **Adjusted Rand Index:** It is another measure of similarity between two data clusterings. It represents the frequency of occurrence of agreements over the total pairs. Its maximum value is 1 which indicates perfect similarity between two clusterings. It is defined as follows,

$$R = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}} \tag{4}$$

where L = contingency Table, $n_{ij} = L[i][j]$, a_i = sum of entries in i^{th} row in L, b_i = Sum of entries in i^{th} column in L, n = number of nodes in the network.

5 Proposed Methodology

Given the computation intractability of the problem statement, we first chunk our approach into two sections. We analyze the structural properties of a small network and generate ground-truth data. This data provides us a way to compare our proposed heuristics, thereby quantifying the effectiveness of these heuristics.

Algorithm 1: Exhaustive Algorithm

```
Input  : Network $G = (V, E)$, $k$, a community detection algorithm $\mathcal{A}$, a value
         function $F$
Output: Set of nodes whose size is $k$
$X \leftarrow$ Run community detection algorithm $\mathcal{A}$ on $G$.
$C \leftarrow$ Generate all the combination of nodes in $V$ of size $k$.
foreach $C_i \in C$ do
    $G' \leftarrow$ Remove $C_i$ from $G$.
    $Y \leftarrow$ Run community detection algorithm $\mathcal{A}$ on $G'$.
    Compute $F$ by comparing $Y$ and $X$.
    Return a $C_i$ which maximizes $F$.
end
```

The thorough approach to gather this information is described in Algorithm 1. This approach compares the networks' community structures before and after structural perturbations, where similarity scores for each combination of nodes are computed.

Yang et al. provide a comparative analysis of significant community detection algorithms, including Edge Betweenness, Fastgreedy, and Infomap. In Algorithm 1, we generate all possible combinations of nodes of size k and then analyze the effect of each such combination to see which minimized the target value function more. Let's consider the computational complexity of the community detection algorithm to be D. This means that the complexity of this algorithm is $O(\max(C(n,k),D))$ where $n = |V|$ and k is the budget. Note that D is generally defined in terms of $|V|$ and $|E|$, so this term becomes more dominant for larger networks, thereby increasing the algorithm's computational complexity.

```
Algorithm 2: Network Based Greedy Approach
  Input  : Network $G = (V, E)$, $k$, a structural property to rank the nodes $P$, a
           community detection algorithm $\mathcal{A}$, a value function $F$
  Output: Set of nodes whose size is $k$, score
  $X \leftarrow$ Run community detection algorithm $\mathcal{A}$ on $G$.
  $R \leftarrow$ Rank the nodes in $G$ based on the structural property $P$.
  $G' \leftarrow$ Remove top $k$ nodes from $G$ based on $R$.
  $Y \leftarrow$ Run community detection algorithm $\mathcal{A}$ on $G'$.
  Compute the value function $F$ by comparing $X$ and $Y$.
  Return the set of top $k$ nodes in $R$ along with the score of the value function
    $F$.
```

Next, we propose a naive network-based greedy approach defined in Algorithm 2. This algorithm takes in a property as an input and ranks the nodes in the input network based on the property specified. It greedily removes the top k nodes based on their ranks and then evaluates the underlying community structure using a community detection algorithm. The output of this algorithm computes the value function and returns the set of nodes removed along with the evaluated value function score. The structural properties which were used are as follows,

- **Clustering Coefficient** - We use the global coefficient, which is defined as the number of closed triplets over the total number of triplets where a triplet is a set of three nodes that are connected by either two or three undirected edges. The complexity of calculating this property for a node is $O(|V|^3)$.
- **Degree Centrality** - It is defined as the number of edges that are incident upon a node. The time complexity of this metric is $O(|V| + |E|)$.
- **Betweenness Centrality** - We use the betweenness centrality estimate defined by Freeman. [30] as the number of times a node acts as a bridge along a shortest path route between two other nodes. Time complexity of this metric is $O(|V||E| + |V|^2)$.
- **Eigenvector Centrality** - It is a measure of the influence of a particular node in the network [11]. This centrality estimate is based on the intuition that a node is more central when there are more connections within its local network. The time complexity of calculating this metric for a node is $O(|V|^3)$.
- **Closeness Centrality** - It measures how easily other vertices can be reached from a particular vertex [9,61]. Time complexity of this metric is $O(|V||E| + |V|^2)$.

- **Coreness** - The coreness of a node is k if it is a member of a k-core but not a member of a $k+1$-core where a k-core is a maximal subnetwork in which each vertex has at least degree k [6]. The time complexity of this metric is $O(|E|)$.
- **Diversity** - The diversity index of a vertex is estimated by the normalized Shannon entropy of the weights of the edges incident on a vertex [26]. The time complexity of calculating this metric is $O(|V| + |E|)$.
- **Eccentricity** - It is defined as the shortest maximum distance from the vertex to all the other vertices in a network. The time complexity of this metric is $O(|V|^2 + |V||E|)$.
- **Constraint** - Introduced by Burt [13], this measure estimates the time and energy that are concentrated on a single cluster. This measure would be higher for a node that belongs to a small network, and also, all the contacts are highly connected. The time complexity of calculating this metric is $O(|V| + |E| + |V|d^2)$ where d is the average degree.
- **Closeness Vitality** - It is defined as the change in the distance between all node pairs when the node in focus is removed. It is based on the Wiener Index, which is defined as the sum of distances between all node pairs [12]. The metric's time complexity is $O(|E| \log |V|)$.

This algorithm takes in a community detection algorithm and a structural property of a node as inputs. If the computational complexity of the community detection algorithm is D and for calculating the structural property for all nodes is S as discussed above, then the total computational complexity of Algorithm 2 is $O(\max(S, D))$.

The downside of this algorithm is that it does not consider the nodes' effect on the community structure of the networks itself. This is addressed in Algorithm 3, which also considers the underlying community structure. Here, we propose a hierarchical approach where we choose a community based on some community-centric metric in the first phase. Then in the second phase, we select a node greedily based on its node-centric properties. The community-centric properties used are as follows:

- **Link density**: $D(G) = \frac{2E}{V(V-1)}$, where E is the number of edges in the network and V is the number of vertices in the network.
- **Conductance**: Given a graph $G(V, E)$, $\lambda(G) = \frac{s}{v}$, where s is number of vertices with one endpoint in G and another in $\bar{G}$, v is the sum of degree of nodes in G. This measure calculates how well-knit a graph is.
- **Compactness**: $C(G)$ is defined as the average shortest path lengths within the network G.

This algorithm takes a community detection algorithm, a node's structural property, and a community-centric property as inputs. The computational complexity of calculating the community-centric property will be constant in time as they are defined in terms of fixed formulas; hence, this would not contribute much to this algorithm's overall complexity. If the computational complexity of the community detection algorithm is D and for calculating the structural

Algorithm 3: Community Based Greedy Approach

```
Input  : Network $G = (V, E)$, $k$, a global community centric property $P_c$, a
         node centric property $P_n$, a community detection algorithm $\mathcal{A}$, a
         value function $F$
Output: Set of nodes whose size if $k$, score
Function best_community(network $G$, community structure $X$):
    foreach $X_i \in X$ do
        $G' \leftarrow$ Create a subnetwork from $G$ with only the vertices from $X_i$.
        $R_g \leftarrow$ Rank each such $G'$ based on the community-centric property $P_c$.
        Return $X_i$ whose induced subnetwork $G'$ ranked above others based on
          $R_g$.
    end
Function best_node(network $G$, community structure $X$):
    $G' \leftarrow$ Create the subnetwork $G'$ from $G$ which is induced from $X$.
    $R_n \leftarrow$ Rank the nodes in $G'$ based on the node centric property $P_n$.
    Return the top node that ranks above others based on $R_n$.
$X \leftarrow$ Run community detection algorithm $\mathcal{A}$ on $G$
$Y \leftarrow$ Run community detection algorithm $\mathcal{A}$ on $G$
while $k$ nodes are not selected do
    $X' \leftarrow$ best_community($G$, $Y$)
    node $\leftarrow$ best_node($G$, $X'$)
    $G' \leftarrow$ Remove node from $G$ and add this node to the output set
    $Y \leftarrow$ Run community detection algorithm $\mathcal{A}$ on $G'$
end
Compute the value function $F$ by comparing $Y$ and $X$.
Return the output set of nodes and the score evaluated by the value function
  $F$.
```

property for all nodes is S as discussed above, then the total computational complexity of Algorithm 2 is $O(\max(S, D))$.

The algorithms proposed above are sufficient for smaller networks. We can evaluate them with Algorithm 1; however, real-world networks exhibit a much more extensive and complex structure.

The reason is the inefficiency of Algorithm 1 as it is a brute force method. This won't allow for the extraction of the ground truth, which we use to estimate the performance of Algorithm 2 and Algorithm 3. To counter this, we propose a new task-based approach. Here, the intuition is as follows: if the performance of an extrinsic task, based on the network structure is ϕ, then after removing the nodes based on the outputs of Algorithm 2 and Algorithm 3, the task would perform $\chi \leq \phi$ on the new network structure, thereby validating the selection of nodes.

Specifically, suppose that a user wants to select vulnerable nodes in an extensive network such that the resulting value function score is maximized. To do so, a straightforward way is to use Algorithm 2 and Algorithm 3 to select the nodes whose effectiveness can be validated by the results of Algorithm 1. But

Algorithm 4: Task Based Approach

```
Input  : Network $G = (V, E)$, $k$, a task $T$, a community detection algorithm
         $\mathcal{A}$, a value function $F$
Output: Set of nodes whose size if $k$, score
Function compute_task_performance(Task $T$, network $G_1$, network $G_2$,
  community structure of $G_1$ $X$, community structure of $G_2$ $Y$):
    if $T$ is link prediction then
        Create a test and train edge list based on the edge set of $G_1$.
        $G'_1 \leftarrow$ Create a subnetwork induced by the training set
        Apply the link prediction task using $X$ to decide on the edge
          probabilities on $G'_1$
        Compute the $F1$ score for the predicted edges
        Repeat the same process for network $G_2$
        Compare the $F1$ scores for both the networks
    end
    else
        Select a random set of seed nodes that are active by default
        With $p_i = 0.7$ and $p_o = 0.3$ apply the information diffusion task on $G_1$
          using the independent cascade model for 200 iterations. This will give
          the number of active nodes at the end of the iterations
        Repeat the process with $G_2$
        Compare the number of active nodes at the end for both $G_1$ and $G_2$
    end
$X \leftarrow$ Run community detection algorithm $\mathcal{A}$ on $G$
$S \leftarrow$ Output from Algorithm 2 or Algorithm 3 which return the target set of
  nodes
$G' \leftarrow$ Remove nodes in $S$ from $G$
$Y \leftarrow$ Run community detection algorithm $\mathcal{A}$ on $G'$
$score \leftarrow$ compute_task_performance($T$, $G$, $G'$, $X$, $Y$)
```

since the network is extensive, it is quite evident that it is not feasible to use Algorithm 1. To counter this, one would use Algorithm 4 to validate the results based on the network's performance drop on the tasks. Since we are using the same algorithms used for small networks, it is evident that the actual problem at hand of maximizing the value function is still of prime focus. Only the way to validate those same results has been changed for more extensive networks.

In Algorithm 4, we consider two different tasks, which are described as follows:

1. **Link Prediction:** We predict the likelihood of a future association between two nodes knowing that there is no association between those nodes in the current state of the network. Hence, the problem asks to what extent the evolution of a complex network can be modeled using features intrinsic to the network topology itself. Generally, in literature, people use few metrics to assign probabilities to a set of non-edges in a network such as Within-

Inter-Cluster defined by Rebaza et al. [63], Modified Common Neighbors and Modified Resource Allocation defined by Soundarajan and Hopcroft [62].

2. **Information Diffusion**: It is defined as the process by which a piece of information is spread and reaches individuals through interactions. We empirically study the behavioral characteristics of information diffusion models, specifically IC (Independent Cascade), on different community structures. We incorporate the community information in this task by assigning p_i probability to edges inside a community and p_o probability to edges that connect separate communities. We keep $p_i \geq p_o$ as information is more likely to spread among nodes within the same neighborhood as observed by Lin et al.[43].

6 Datasets

To run our experiments extensively, we select six real-world networks of diverse sizes. The datasets used are as follows:

1. **Karate Club:** The data was collected from the members of a karate club [70,69]. Each node represents a club member, and each undirected edge represents a tie between two members of the club. The network has two communities, one formed by ”John A” and another by ”Mr Hi”.
2. **Football Network:** Girvan and Newman [31] collected this network. It contains American football games between division IA colleges during the regular season Fall of 2000. The nodes represent teams identified by names, and edges represent regular-season games between two teams that they connect. The network has twelve communities where each community is signified by the conferences that each college belongs to.
3. **Indian Railway Network:** This network was used in [20], which consists of nodes that represent stations where two stations are connected by an edge if there exists at least one train route between them such that these stations are scheduled halts. The states act as communities, and hence there are 21 communities.
4. **Co-authorship Network:** This network was collected by Chakraborty et al. [20]. This dataset comprises nodes representing an author, and an undirected edge between two authors is drawn if and only if they were co-authors at least once. Each author is tagged with one research field on which he/she has written most papers on. There are 24 such fields, and they act as communities.
5. **Amazon Product Co-purchasing Network:** This was collected by crawling the Amazon site [67]. The nodes represent products, and an undirected edge between two nodes represents a frequently co-purchased product. There are 75,149 communities, and only groups containing more than three users are considered.
6. **Live Journal:** This is a free online blogging community where users declare friendship with each other [67]. Therefore, each node is a user, and an edge between two users represents a friendship. Users are allowed to form groups,

and such user-defined groups form communities. There are 287,512 communities, and only groups containing more than three users are considered.

Table 1: Properties of the real-world networks used in our experiments. We chose 3 small and 3 large networks to extensively show the effects of each algorithm in terms of efficiently computing the vulnerable communities.

Dataset	**#Nodes**	**#Edges**	**#Communities**
Karate Club Network	34	78	2
Football Network	115	613	12
Indian Railway Network	301	1,224	21
Co-authorship Network	103,667	352,183	24
Amazon Product Co-purchasing Network	334,863	925,872	75,149
Live Journal Network	3,997,962	34,681,189	287,512

7 Experiments

We divide this section into three subsections to cover all the value functions discussed in Section 4. We first present the results of Algorithm 1 for smaller networks, which will be used as a benchmark to compare the results of Algorithm 2 and Algorithm 3 whose results will follow. Using the inferences from these results, we build on our argument and present the results of Algorithm 4 to establish similar results even on more extensive networks.

7.1 Modularity

Exhaustive approach: Table 2 shows the results of Algorithm 1 on three small networks when using the modularity as the target value function. We perform the analysis by fixing $k = 5$ [1]. For the Karate network, we observe that nodes (0, 1, 3, 5, 6) are the most vulnerable as their removal maximizes the difference of modularity scores between the original and the perturbed networks. Similarly, the most susceptible nodes identified for the other two networks are mentioned in Table 2.

[1] We choose the value of k to be five because of the following reason. Since we intend to compare our approach with the ground truth data, we first need to generate this ground truth data. For smaller k values, the number of nodes' combinations is more diminutive and keeps increasing exponentially as we increase the value of k. To limit the computational time, we restricted k to be five, and beyond that, the number of combination of nodes was too large. Simultaneously, we did not want to choose a smaller k, as removing a smaller number of nodes would not have that much impact on the underlying community structure than removing more nodes.

Table 2: Effect of the exhaustive algorithm on the small networks. The nodes here indicate the ID of the most vulnerable points in the network when the modularity is utilized as the value function. Since the networks are smaller in size, the budget k is fixed at 5 which is why the algorithm detects only 5 vulnerable nodes. The corresponding modularity score reported is the maximum across all possible combinations of the nodes.

Network	Nodes	Modularity
Karate	(0, 1, 3, 5, 6)	0.13436
Football	(23, 33, 24, 32, 45)	0.10492
Railway	(105, 76, 203, 123, 97)	0.14723

Network Based Greedy Approach: This section presents the analysis results on all the datasets of Algorithm 2. We performed this analysis on all the datasets irrespective of their scale as the algorithm applied was greedy and did not need much time to execute. Moreover, we fix $k = 5$ for smaller networks, but such a removal strategy won't showcase significant effects for more extensive networks. This is because removing just five nodes in more extensive networks won't affect the underlying community structure enough to cause substantial structural perturbations. So, to handle such cases, we instead remove 5% of the total nodes. From Figure 1, we infer that the clustering coefficient as a network-based greedy metric performs better than other greedy metrics when we remove the target five nodes.

Moreover, when we compare the maximum values attained in the smaller networks, we see that this algorithm cannot achieve the optimal answer indicated in Table 2. For example, in the Karate network, the maximum score obtained by Algorithm 2 is around 0.05, whereas the optimal answer is 0.13. This indicates that there is a lot of scope for improvement.

Community Based Greedy Approach: In this section, we evaluate the performance of Algorithm 3 over all the datasets. As mentioned previously, we fix $k = 5$ for smaller networks and 5% for more extensive networks. We compare the different community-centric properties in Table 3. Here we present the best modularity scores obtained after applying this algorithm on all the datasets. As this algorithm is also inherently greedy, it also is computationally efficient. Based on Table 3, we observe that Link Density performed better than the other community-centric properties as the scores over all the datasets were maximum. Now that we have established that the best community-centric property in a modularity difference maximization setting is link density, we present the node centric properties' results in Figure 2. Overall, we ran experiments on the datasets; we found that eigenvector centrality performs better than other greedy metrics.

Additionally, when we compare this algorithm's results with the ground truth data presented in Table 2, we observe that this solution comes close to the

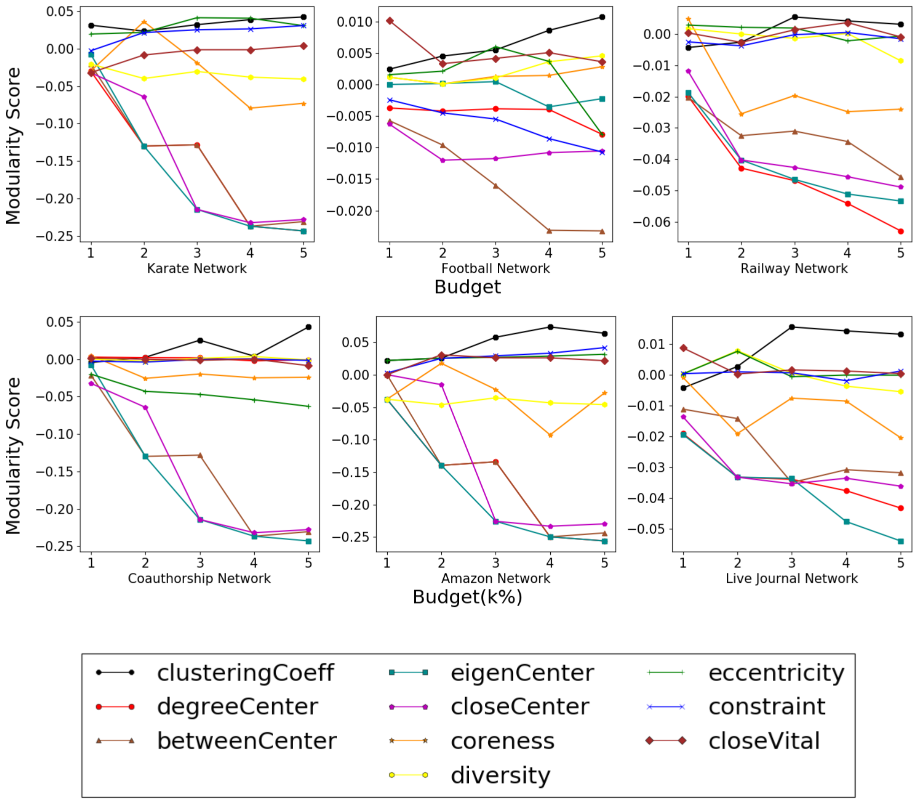

Fig. 1: Outcome of the network based approach over all the networks with modularity being the target value function. The legend indicates all the structural properties of a network that were used to greedily select nodes. For smaller networks we used $k = 5$ nodes whereas for larger network s we used 5% of the nodes in the corresponding network. If we compare the smaller datasets' results with Table 2, we observe that the maximum values obtained could not attain the optimal values. Across the networks, for higher budget we observe that clustering coefficient turned out to be the best indicator for vulnerability in terms of modularity.

optimal solution. For example, in the Railway network, we follow that the best modularity score obtained to be around 0.06, which is close to the ground truth score of 0.14 compared to the 0.01 score obtained from Algorithm 1. So it is evident from this data that the difference between the optimal solution and the current solution has decreased, thereby establishing the superiority of Algorithm 3 over 2.

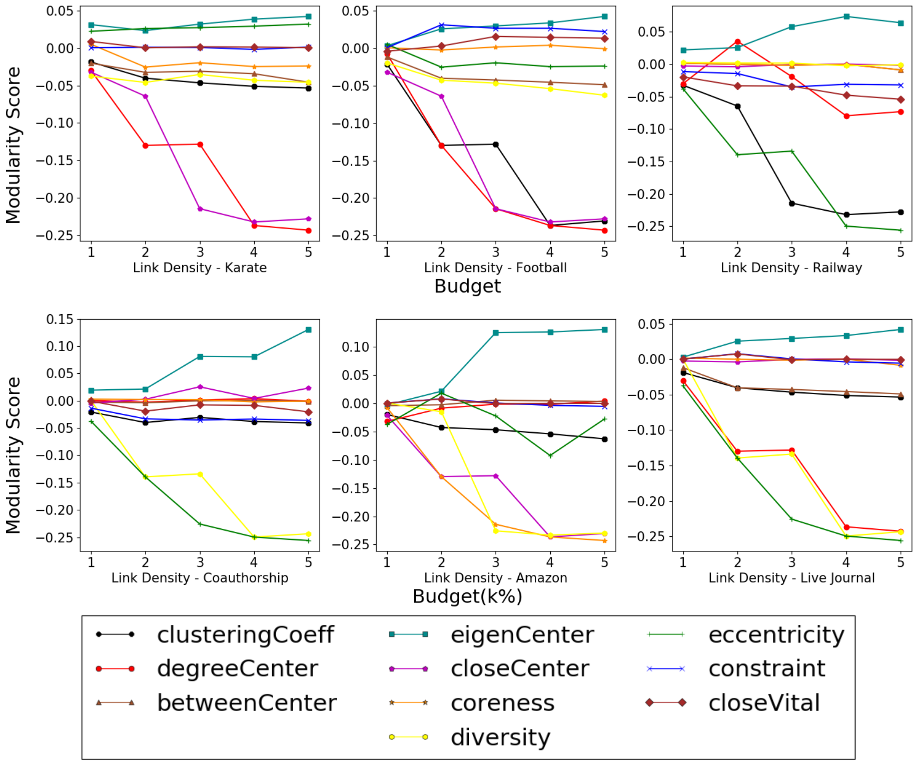

Fig. 2: Results of the community based approach over several datasets with modularity being used as the target value function. These results are reported only for Link Density as it outperformed the other community based greedy metrics as described in Table 3. The plots indicate that across the datasets for larger budgets, eigenvector centrality performs better in comparison to other greedy metrics. The values indicated are close to the ground truth as reported in Table 1.

7.2 Normalized Mutual Information

Exhaustive approach: Table 4 presents the results of Algorithm 1 on three small scale datasets where the value function that we are trying to minimize is

Table 3: Outcome of the community based approach using modularity as the target value function. It shows the effects of different community based metrics have when used to greedily select nodes. Based on this table we observe that Link Density performs better to indicate the vulnerability of nodes in terms of difference between modularity of the resultant and the original network across all the datasets.

Network	**Link Density**	**Conductance**	**Compactness**
Karate	0.04194	0.00116	0.02219
Football	0.04202	0.02193	0.00490
Railway	0.06422	0.03174	0.03749
Coauthorship	0.13037	0.03609	0.00285
Amazon	0.13052	0.00550	0.01783
Live Journal	0.05289	0.00016	0.03749

NMI. Note that we would want to minimize NMI as this metric gives a value of 1 for two similar community structures and 0 otherwise as mentioned in Section 4. For this experiment, we fix the number of target nodes, i.e., $k = 5$. For the football network, we observe that nodes (32, 33, 5, 6, 1) are identified as the most vulnerable as they minimize the NMI score between the original and the structurally perturbed one to 0.38. This value represents the ground truth as no other combination of the five-tuple nodes will further decrease the NMI score between the two partitions. Similarly, the other small datasets' ground truth values can be found in Table 4.

Table 4: Effect of the exhaustive algorithm on the smaller networks. The nodes here indicate the ID of the most vulnerable points in the network when NMI is utilized as the value function. Since the networks are smaller in size the budget k was fixed at 5 which is why there the algorithm detected 5 vulnerable nodes. The corresponding NMI score reported was the minimum across all possible combinations of the nodes.

Network	**Nodes**	**NMI**
Karate	(33, 10, 32, 6, 23)	0.36762
Football	(32, 33, 5, 6, 1)	0.38580
Railway	(51, 143, 2, 89, 287)	0.38723

Network Based Greedy Approach: This section presents the analysis results on all the datasets of Algorithm 2. Moreover, we fix $k = 5$ for smaller networks, as mentioned previously. Still, for more extensive networks, such removal strategy won't showcase significant effects, and hence we remove till 5% of the total nodes in such cases. Based on Figure 3, we infer that eccentricity as a network-based greedy metric performs better than other greedy metrics when we remove the target five nodes. As we evaluate the NMI measure, we compare the minimum values attained in the ground truth data to the minimum values obtained with Algorithm 2. This is because NMI's value is small when two clusterings are not the same as mentioned previously in Section 4. Based on this comparison for smaller networks, we see that this algorithm could not attain the optimal answer indicated by Table 4. For example, in the Karate network, the minimum score obtained by Algorithm 2 is 0.55, whereas the optimal answer is 0.36. This indicates that there is a lot of scope for improvement.

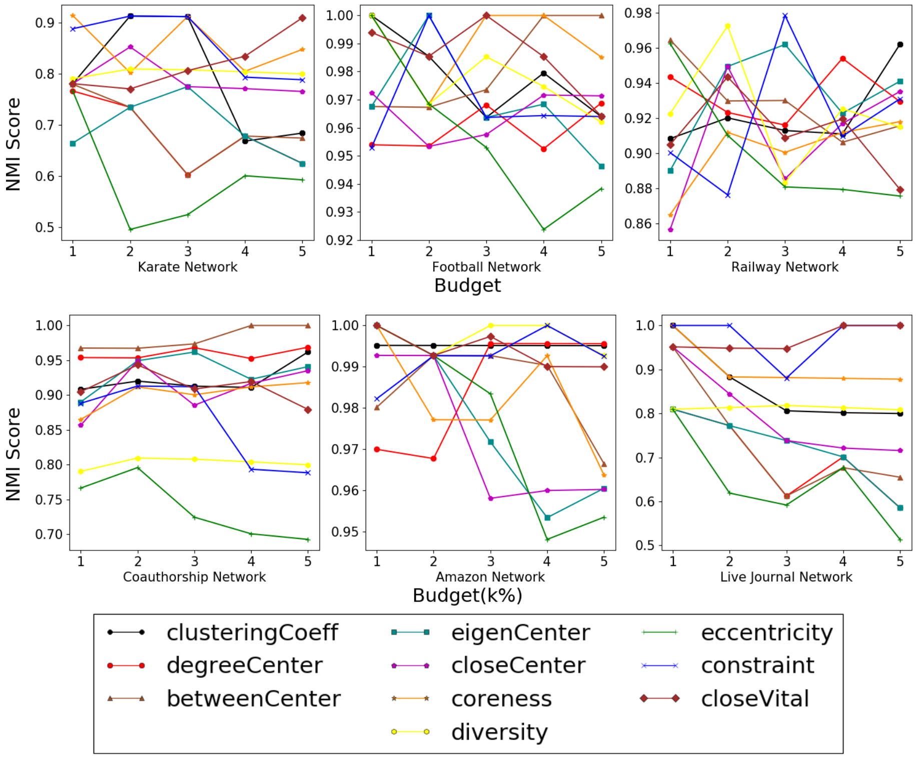

Fig. 3: Results of the network-based approach over all the datasets with NMI being the target value function. For smaller networks we use $k = 5$ and for larger ones we use 5% of the total nodes within the network. The plots indicate that for larger budgets, eccentricity performs well in identifying vulnerable nodes when the vulnerability of a community is quantified using NMI where lower values are better indicators of disjointness. However, upon closer inspection one can observe that these values when compared to the ground truth values reported in Table 4, are still pretty far from optimal.

Community Based Greedy Approach: In this section, we evaluate the performance of Algorithm 3 over all the datasets. As mentioned previously, we fix

$k = 5$ for smaller networks and 5% for more extensive networks. We compare the different community-centric properties in Table 5. Here we present the best NMI scores obtained after applying this algorithm on all the datasets. Based on Table 5, we observe that Link Density performed better than the other community-centric properties as the scores over all the datasets were minimal. With link density as the best community-centric method, we present the node centric properties' results in Figure 4. Overall the datasets we ran experiments on, we found that the clustering coefficient performs better than other greedy metrics.

Additionally, when we compare this algorithm's results with the ground truth data presented in Table 4, we observe that this solution comes close to the optimal solution. For example, in the Railway network, we follow that the best NMI score obtained to be around 0.5 is close to the ground truth score of 0.38 compared to the 0.88 score obtained from Algorithm 1. So it is evident from this data that the difference between the optimal solution and the current solution has decreased, thereby establishing the superiority of Algorithm 3 over 2.

Table 5: Results of the community based approach using NMI as the target value function. It shows the effects of different community based metrics that are used to greedily select nodes. We observe that Link Density performs better to indicate the vulnerability of nodes in terms of the NMI between the resultant and the original network across all the datasets.

Network	**Link Density**	**Conductance**	**Compactness**
Karate	0.62484	0.68425	0.79993
Football	0.75558	0.96877	0.91794
Railway	0.51372	0.80825	0.62484
Coauthorship	0.59279	0.71568	0.79993
Amazon	0.58566	0.76560	0.65510
Live Journal	0.58566	0.62484	0.78850

7.3 Adjusted Rand Index

Exhaustive approach: Table 6 shows the results of Algorithm 1 on three small scale datasets when using the ARI as the target value function. We performed the analysis by fixing $k = 5$. We observe that nodes (61, 85, 16, 99, 7) are the most vulnerable for the football network as their removal minimized the ARI scores between the original and the perturbed network's vertex clusterings. Similarly, the most susceptible nodes identified for the other two datasets have been tabulated in Table 6.

Network Based Greedy Approach: This section presents the analysis results on all the datasets of Algorithm 2. We fix $k = 5$ for smaller networks as mentioned previously, but for more extensive networks, we remove till 5% of the

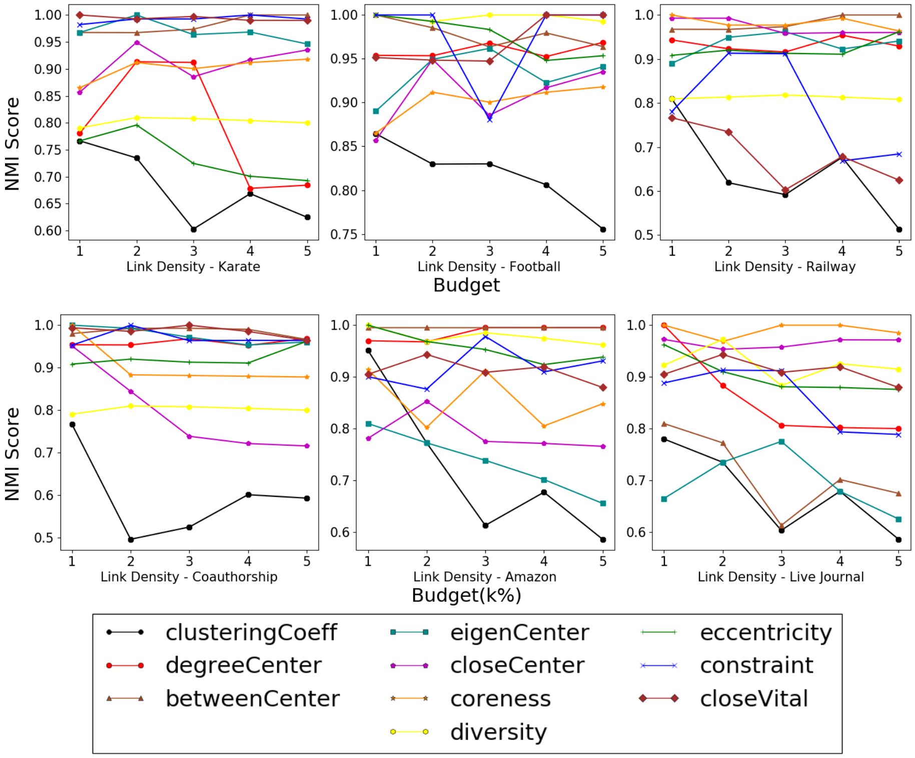

Fig. 4: Outcome of the community based approach over all the datasets with NMI being the target value function. Based on Table 5 we observed that Link Density performs better in comparison to other greedy metric. The plots reported in this figure present the results of the node centric properties with Link Density as the community centric method. They indicate that across all the datasets clustering coefficient performed better compared to other greedy metrics.

Table 6: Effect of the exhaustive algorithm on the smaller networks. The nodes here indicate the ID of the most vulnerable points in the network when ARI is utilized as the value function. Since the networks are smaller in size the budget k was fixed at 5 which is why there the algorithm detected 5 vulnerable nodes. The corresponding ARI score reported was the minimum across all possible combinations of the nodes.

Network	Nodes	ARI
Karate	(32, 7, 12, 18, 2)	-0.46342
Football	(61, 85, 16, 99, 7)	0.36342
Railway	(171, 229, 236, 75, 204)	-0.28694

total nodes. Based on Figure 5, we infer that closeness vitality as a network-based greedy metric performs better than other greedy metrics when we remove the target five nodes. As we evaluate the ARI measure, we compare the minimum values attained in the ground truth data to the minimum values obtained with Algorithm 2. This is because ARI's value is small when two clusterings do not agree with each other, as mentioned previously in Section 4. Based on this comparison for smaller networks, we see that this algorithm cannot attain the optimal answer mentioned in Table 6. For example, in the Railway network, the minimum score obtained by Algorithm 2 is 0.65, whereas the optimal answer is -0.28. This indicates that there is a lot of scope for improvement.

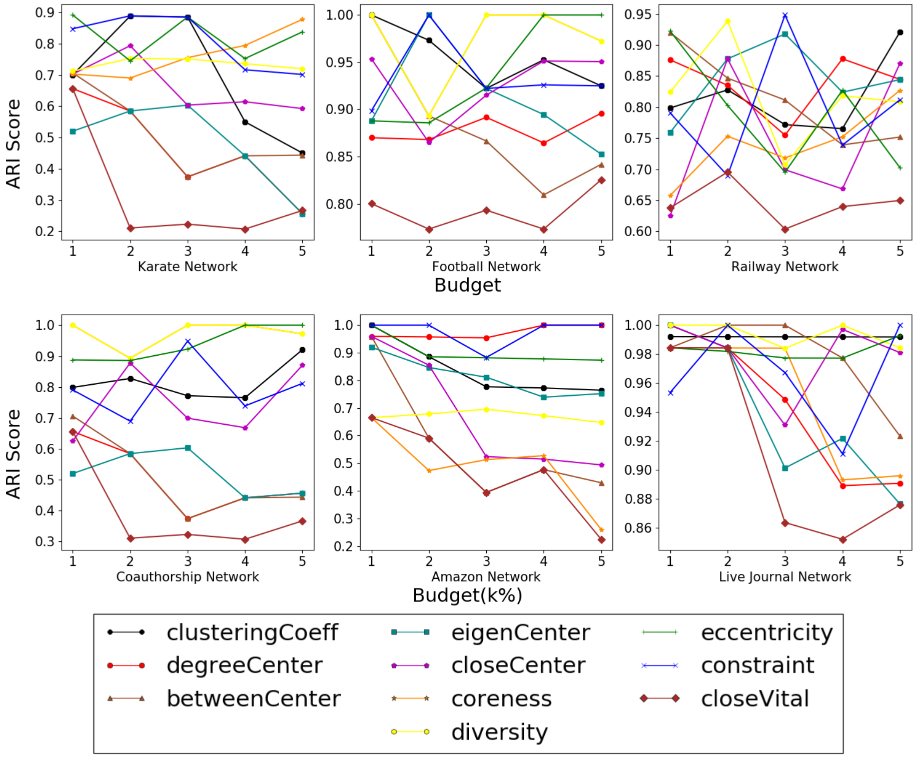

Fig. 5: Results of the network based approach applied on several datasets with ARI being the target value function. We chose $k = 5$ for smaller networks and for larger networks we chose upto 5% of the total nodes within the network. The plots reported in this figure show that closeness vitality performed better than the other node centric properties as for larger budgets the ARI between the resultant and the original network was low. When compared to the exhaustive results as reported in Table 6, we observe that the ARI values with closeness vitality as the greedy metric does not come close to the optimal answer.

Community Based Greedy Approach: In this section, we evaluate the performance of Algorithm 3 over all the datasets. As mentioned previously, we fix $k = 5$ for smaller networks and 5% for larger networks. We compare different community-centric properties in Table 7. Here we present the best ARI scores obtained after applying this algorithm on all the datasets. We observe that conductance performs better than the other community-centric properties as the scores over all the datasets are minimum. With conductance as the best community-centric method, we present the node-centric properties' results in Figure 6. Overall the datasets we run experiments on, we find that coreness performs better compared to other metrics.

Additionally, when we compare this algorithm's results with the ground truth data presented in Table 6, we observe that this solution comes close to the optimal solution. For example, in the Railway network, we follow that the best ARI score obtained to be around 0.26 is close to the ground truth score of -0.28 compared to the 0.65 score obtained from Algorithm 1. So it is evident from this data that the difference between the optimal solution and the current solution has decreased, thereby establishing the superiority of Algorithm 3 over Algorithm 2.

Table 7: Results of the community based approach using ARI as the target value function. It shows the effects of different community based metrics used to greedily select nodes. We observe that conductance performs better to indicate the vulnerability of nodes in terms of the ARI between the resultant and the original community structure across all the datasets.

Network	**Link Density**	**Conductance**	**Compactness**
Karate	0.45034	0.25670	0.82691
Football	0.82530	0.64736	0.89587
Railway	0.44367	0.26693	0.27997
Coauthorship	0.71958	0.45670	0.75187
Amazon	0.69230	0.64979	0.76453
Live Journal	0.44367	0.25670	0.26693

7.4 Task Based Approach

Based on the results that we observed in the previous sections for the smaller networks, we perform similar tests on more extensive networks using Algorithm 4. To quantify this algorithm's performance, we use the widely use F1 score for the link prediction task. We evaluate the fraction of active nodes at the end of the few cascades for the information diffusion task. For each experiment, we consider k to be the percentage of nodes removed as otherwise the change in the community structure would not be enough to have significant effects. We have divided this section into two subsections to cover both the tasks that were described before.

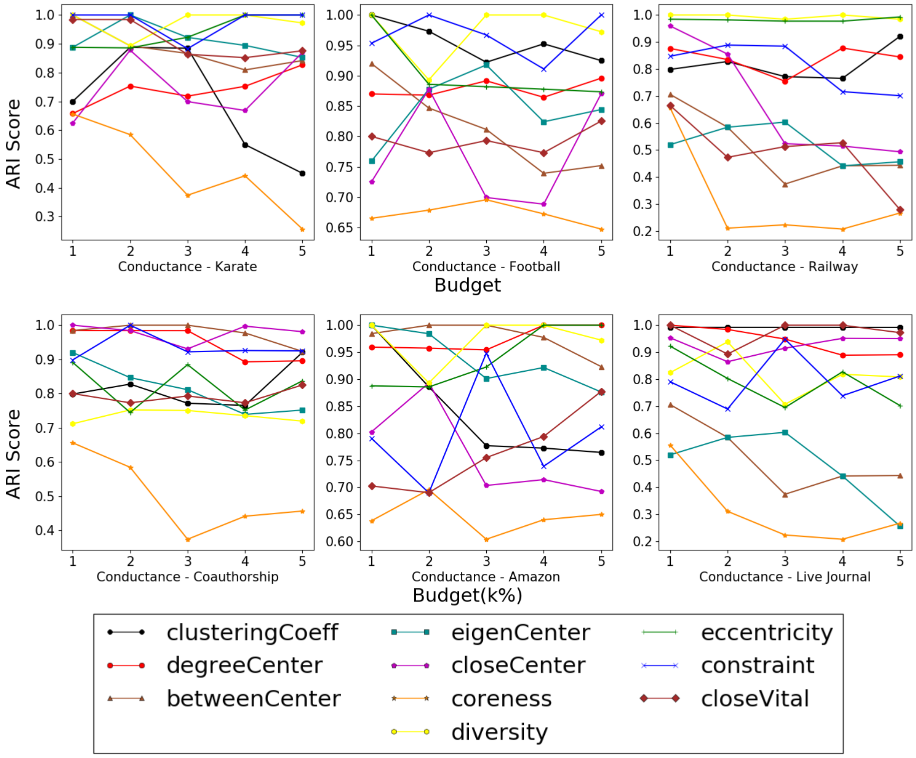

Fig. 6: Outcome of the community based approach over all the datasets with ARI being the target value function. Based on Table 7, we observe that conductance performs better compared to other community based methods to quantify the vulnerability of communities using ARI. The plots show the effects of different node-centric properties with conductance in Algorithm 3. The results show that across all the datasets, coreness outperforms other metrics. Upon comparing these results with the ground truth data in Table 6, we observe that the values are close to the optimal answers.

Link Prediction: We test this task by assigning probabilities to the edges using three metrics separately: Within-Inter Cluster, Modified Common Neighbors, and Modified Resource Allocation. We find that Within-Inter Cluster produces better results compared to the other alternatives.

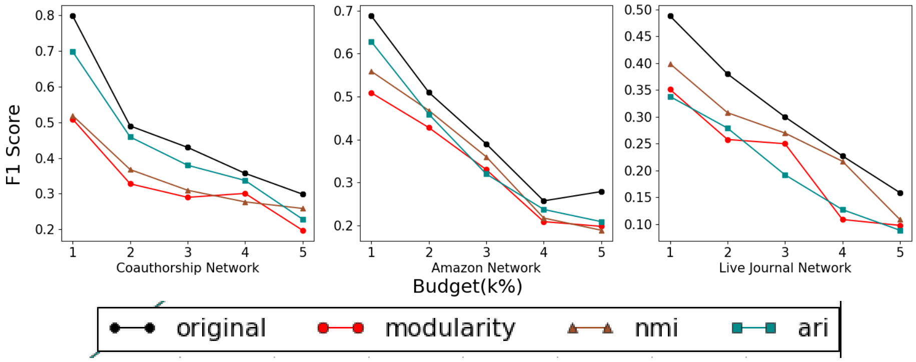

Fig. 7: Results of link prediction task over larger datasets with all the value functions. For modularity, we use Link Density combined with eigenvector centrality. For NMI, we use Link Density combined with eccentricity, and for ARI, we use conductance and closeness vitality. Original here represents when we use the original community structure for the nodes rather than the community structure after perturbing it. The plots indicate that when using these combinations for different value functions, the link prediction task's performance quantified with the F1 score decreases.

Based on Figure 7, we observe that overall value functions the network's performance in the link prediction task has decreased, which is evident from the lower F1 scores. For each value function, we show the best combination as identified in the previous sections. The performance drop can be attributed to significant changes introduced into the system by removing vulnerable nodes. Their removal triggers significant structural perturbations in the underlying community structure, which causes the within-inter cluster method to assign lower probabilities to the edges due to fewer connections within the community and more connections across other communities. This decreased the likelihood of the test edge being classified as a valid link, thereby reducing the performance.

Information Diffusion: In 8, we observe that overall value functions the performance in the information diffusion task has decreased, which is evident from the lower fraction of active nodes. For this set of experiments, we set $p_i \geq p_o$ and let the cascade model run for 200 iterations. With a higher probability for the initial set and the subsequent set of active nodes to affect the nodes within their community, it is trivial to see that the fraction of nodes that will be active at the end of all the iterations would be low. This is true if the underlying community structure was significantly perturbed and the network was highly disconnected, whereas it would be the opposite for the other case. For each value function, we show the best combination as identified in the previous sections.

This shows that the best combination of community-centric and network-centric nodes that we get from Algorithm 3 when applied to the more extensive networks using Algorithm 4 results in the decrease in the performance of the networks over both tasks that they are employed on, thereby validating our initial hypothesis. This establishes that Algorithm 3 can be applied to any general network irrespective of the size.

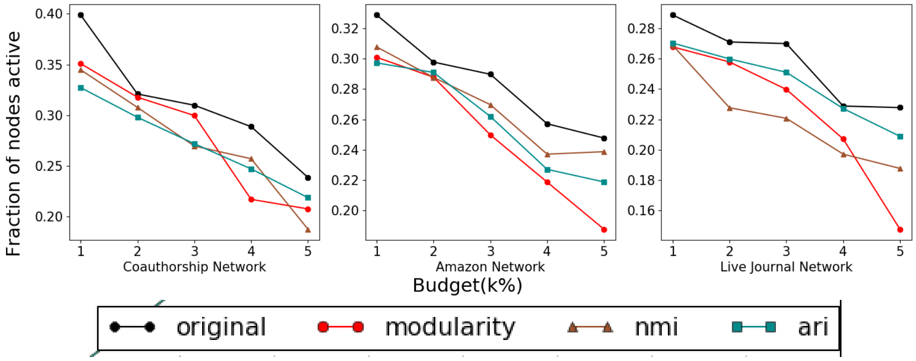

Fig. 8: Outcome of information diffusion task over larger networks with all the value functions. For modularity we use Link Density along with eigenvector centrality, for NMI we use Link Prediction combined with eccentricity and for ARI we utilize conductance combined with closeness vitality. Original here represents when we use the original community structure for the nodes rather than the community structure after perturbing it. The plots show that for all the target value functions the performance of the information diffusion task decreases. This performance was quantified using the fraction of *active* nodes after all the iterations.

8 Conclusion

In this paper, we proposed a hierarchical greedy-based approach that efficiently identified critical nodes in the network, which significantly impacted the underlying community structure. Additionally, we also proposed a novel task-based strategy to apply the results of the hierarchical greedy based approach on more extensive networks and quantify its effectiveness, which would enable us to estimate the performance of the algorithm in a real-world context.

Due to the extensive size of our experiments, we show our best results only. Since Algorithm 1 is exhaustive and hence was applied only to small networks such as Karate, Football and Railway Network. The results of this algorithm provided us with the benchmark to compare with our other algorithms. We further saw that Algorithm 2 was not that promising and were far from the gold standard in comparison to Algorithm 3 which came close to the gold standard. This comparison showed that Algorithm 3 works best for small networks. As mentioned previously, we used Algorithm 4 to compare the performance of Algorithm 2 and Algorithm 3 for large networks such as Co-authorship, Amazon, and Live Journal Networks. Based on these results, we established that when we use Algorithm 3, we get a performance drop over both the tasks, namely link prediction and information diffusion, compared to the original network. This establishes the generalizability of Algorithm 3.

To conclude, this work has provided a hierarchical approach that allowed for identifying the vulnerable nodes in a network efficiently. The proposed method was used to analyze the community vulnerability of several networks whose validity was established using both exhaustive and task-based approaches depending on the network's size.

Acknowledgement. T. Chakraborty would like to acknowledge the support of SERB (ECR/2017/001691) and the Infosys Centre for AI, IIIT-Delhi.

References

1. Adamcsek, B., Palla, G., Farkas, I.J., Derenyi, I., Vicsek, T.: CFinder: locating cliques and overlapping modules in biological networks. Bioinformatics **22**(8), 1021–1023 (02 2006). https://doi.org/10.1093/bioinformatics/btl039
2. Agarwal, P., Verma, R., Agarwal, A., Chakraborty, T.: Dyperm: Maximizing permanence for dynamic community detection. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 437–449. Springer (2018)
3. Alim, M.A., Li, X., Nguyen, N.P., Thai, M.T., Helal, A.: Structural vulnerability assessment of community-based routing in opportunistic networks. IEEE Transactions on Mobile Computing **15**(12), 3156–3170 (Dec 2016). https://doi.org/10.1109/TMC.2016.2524571
4. Allesina, S., Pascual, M.: Googling food webs: Can an eigenvector measure species' importance for coextinctions? PLOS Computational Biology **5**(9), 1–6 (09 2009). https://doi.org/10.1371/journal.pcbi.1000494
5. Bader, D.A., Meyerhenke, H., Sanders, P., Wagner, D. (eds.): Graph Partitioning and Graph Clustering, 10th DIMACS Implementation Challenge Workshop, Georgia Institute of Technology, Atlanta, GA, USA, February 13-14, 2012. Proceedings, Contemporary Mathematics, vol. 588. American Mathematical Society (2013). https://doi.org/10.1090/conm/588
6. Batagelj, V., Zaversnik, M.: An o(m) algorithm for cores decomposition of networks. CoRR **cs.DS/0310049** (2003), `http://arxiv.org/abs/cs.DS/0310049`
7. Baumes, J., Goldberg, M., Magdon-Ismail, M.: Efficient identification of overlapping communities. In: Kantor, P., Muresan, G., Roberts, F., Zeng, D.D., Wang, F.Y., Chen, H., Merkle, R.C. (eds.) Intelligence and Security Informatics. pp. 27–36. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
8. Baumes, J., Goldberg, M.K., Krishnamoorthy, M.S., Magdon-Ismail, M., Preston, N.: Finding communities by clustering a graph into overlapping subgraphs. IADIS AC **5**, 97–104 (2005)
9. Bavelas, A.: Communication patterns in task-oriented groups. Acoustical Society of America Journal **22**, 725 (1950). https://doi.org/10.1121/1.1906679

10. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10), P10008 (2008), `http://stacks.iop.org/1742-5468/2008/i=10/a=P10008`
11. Bonacich, P.: Factoring and weighting approaches to status scores and clique identification. The Journal of Mathematical Sociology **2**(1), 113–120 (1972). https://doi.org/10.1080/0022250X.1972.9989806
12. Brandes, U.: Network analysis: methodological foundations, vol. 3418. Springer Science & Business Media (2005)
13. Burt, R.: Structural holes and good ideas. American Journal of Sociology **110**(2), 349–399 (2004), `http://www.jstor.org/stable/10.1086/421787`
14. Chakraborty, T.: Leveraging disjoint communities for detecting overlapping community structure. Journal of Statistical Mechanics: Theory and Experiment **2015**(5), P05017 (2015)
15. Chakraborty, T., Dalmia, A., Mukherjee, A., Ganguly, N.: Metrics for community analysis: A survey. ACM Computing Surveys (CSUR) **50**(4), 54 (2017)
16. Chakraborty, T., Ghosh, S., Park, N.: Ensemble-based overlapping community detection using disjoint community structures. Knowledge-Based Systems **163**, 241–251 (2019)
17. Chakraborty, T., Kumar, S., Ganguly, N., Mukherjee, A., Bhowmick, S.: Genperm: a unified method for detecting non-overlapping and overlapping communities. IEEE Transactions on knowledge and data engineering **28**(8), 2101–2114 (2016)
18. Chakraborty, T., Park, N.: Ensemble-based discovery of disjoint, overlapping and fuzzy community structures in networks. arXiv preprint arXiv:1712.02370 (2017)
19. Chakraborty, T., Park, N., Subrahmanian, V.: Ensemble-based algorithms to detect disjoint and overlapping communities in networks. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 73–80. IEEE (2016)
20. Chakraborty, T., Srinivasan, S., Ganguly, N., Mukherjee, A., Bhowmick, S.: On the permanence of vertices in network communities. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1396–1405. KDD '14, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2623330.2623707
21. Chan, H., Akoglu, L., Tong, H.: Make it or break it: Manipulating robustness in large networks. In: Proceedings of the 2014 SIAM International Conference on Data Mining. pp. 325–333. SIAM (2014)
22. Chen, W., Liu, Z., Sun, X., Wang, Y.: A game-theoretic framework to identify overlapping communities in social networks. Data Min. Knowl. Discov. **21**, 224–240 (09 2010). https://doi.org/10.1007/s10618-010-0186-6
23. Clauset, A., Newman, M.E.J., , Moore, C.: Finding community structure in very large networks. Physical Review E pp. 1– 6 (2004). https://doi.org/10.1103/PhysRevE.70.066111, `www.ece.unm.edu/ifis/papers/community-moore.pdf`
24. Danon, L., Díaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment **2005**(09), P09008 (2005), `http://stacks.iop.org/1742-5468/2005/i=09/a=P09008`
25. De Meo, P., Ferrara, E., Fiumara, G., Provetti, A.: Enhancing community detection using a network weighting strategy. Inf. Sci. **222**, 648–668 (Feb 2013). https://doi.org/10.1016/j.ins.2012.08.001

26. Eagle, N., Macy, M., Claxton, R.: Network diversity and economic development. Science **328 5981**, 1029–31 (2010)
27. Farkas, I., Ábel, D., Palla, G., Vicsek, T.: Weighted network modules. New Journal of Physics **9**(6), 180–180 (jun 2007). https://doi.org/10.1088/1367-2630/9/6/180
28. Fiedler, M.: Algebraic connectivity of graphs. Czechoslovak Mathematical Journal **23**(2), 298–305 (1973), `http://eudml.org/doc/12723`
29. Frank, H., Frisch, I.: Analysis and design of survivable networks. IEEE Transactions on Communication Technology **18**(5), 501–519 (October 1970). https://doi.org/10.1109/TCOM.1970.1090419
30. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry **40**(1), 35–41 (1977), `http://www.jstor.org/stable/3033543`
31. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences **99**(12), 7821–7826 (2002). https://doi.org/10.1073/pnas.122653799
32. Grubesic, T.H., Matisziw, T.C., Murray, A.T., Snediker, D.: Comparative approaches for assessing network vulnerability. International Regional Science Review **31**(1), 88–112 (2008). https://doi.org/10.1177/0160017607308679
33. Guimera, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. Nature **433**(7028), 895–900 (feb 2005), `http://dx.doi.org/10.1038/nature03288`
34. Havemann, F., Heinz, M., Struck, A., Gläser, J.: Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels. Journal of Statistical Mechanics: Theory and Experiment **2011**(01), P01023 (jan 2011). https://doi.org/10.1088/1742-5468/2011/01/p01023
35. Holme, P., Kim, B.J., Yoon, C.N., Han, S.K.: Attack vulnerability of complex networks. Phys. Rev. E **65**, 056109 (May 2002). https://doi.org/10.1103/PhysRevE.65.056109
36. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification **2**, 193–218 (1985). https://doi.org/10.1007/BF01908075
37. Kanawati, R.: Yasca: an ensemble-based approach for community detection in complex networks. In: International Computing and Combinatorics Conference. pp. 657–666. Springer (2014)
38. Lancichinetti, A., Fortunato, S.: Community detection algorithms: A comparative analysis. Physical Review **80**(5) (2009). https://doi.org/10.1103/PhysRevE.80.056117
39. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. New Journal of Physics **11**(3), 033015 (mar 2009). https://doi.org/10.1088/1367-2630/11/3/033015
40. Lancichinetti, A., Radicchi, F., Ramasco, J.J., Fortunato, S.: Finding statistically significant communities in networks. PLOS ONE **6**(4), 1–18 (04 2011). https://doi.org/10.1371/journal.pone.0018961
41. Lee, C., Reid, F., McDaid, A., Hurley, N.: Detecting highly overlapping community structure by greedy clique expansion. KDD SNA 2010 (02 2010)
42. Li, P., Salour, M., Su, X.: A survey of internet worm detection and containment. IEEE Communications Surveys Tutorials **10**(1), 20–35 (First 2008). https://doi.org/10.1109/COMST.2008.4483668
43. Lin, S., Hu, Q., Wang, G., Yu, P.S.: Understanding community effects on information diffusion. In: Cao, T., Lim, E.P., Zhou, Z.H., Ho, T.B., Cheung, D., Motoda, H. (eds.) Advances in Knowledge Discovery and Data Mining. pp. 82–95. Springer International Publishing, Cham (2015)

44. Nepusz, T., Petróczi, A., Négyessy, L., Bazsó, F.: Fuzzy communities and the concept of bridgeness in complex networks. Physical review. E, Statistical, nonlinear, and soft matter physics **77 1 Pt 2**, 016107 (2008)
45. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences **103**(23), 8577–8582 (2006). https://doi.org/10.1073/pnas.0601602103
46. Newman, M.E.J.: Community detection and graph partitioning. EPL (Europhysics Letters) **103**(2), 28003 (jul 2013). https://doi.org/10.1209/0295-5075/103/28003
47. Newman, M.E.J., Leicht, E.A.: Mixture models and exploratory analysis in networks. Proceedings of the National Academy of Sciences **104**(23), 9564–9569 (2007). https://doi.org/10.1073/pnas.0610537104
48. Newman, M.: Fast algorithm for detecting community structure in networks. Physical Review E **69** (September 2003), `http://arxiv.org/abs/cond-mat/0309508`
49. Nguyen, H.T., Nguyen, N.P., Vu, T., Hoang, H.X., Dinh, T.N.: Transitivity demolition and the fall of social networks. IEEE Access **5**, 15913–15926 (2017). https://doi.org/10.1109/ACCESS.2017.2672666
50. Nguyen, N.P., Alim, A., Shen, Y., Thai, M.T.: Assessing network vulnerability in a community structure point of view. 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013) pp. 231–235 (2013). https://doi.org/10.1145/2492517.2492644
51. Nowicki, K., Snijders, T.A.B.: Estimation and prediction for stochastic blockstructures. Journal of the American Statistical Association **96**(455), 1077–1087 (2001). https://doi.org/10.1198/016214501753208735
52. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**(7043), 814–818 (June 2005), `http://dx.doi.org/10.1038/nature03607`
53. Raghavan, U., Albert, R., Tirupatikumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Physical Review E - Statistical, Nonlinear, and Soft Matter Physics **76**(3) (9 2007). https://doi.org/10.1103/PhysRevE.76.036106
54. Ramirez-Marquez, J.E., Rocco, C.M., Barker, K., Moronta, J.: Quantifying the resilience of community structures in networks. Reliability Engineering and System Safety **169**, 466–474 (2018), `https://doi.org/10.1016/j.ress.2017.09.019`
55. Ren, W., Yan, G., Liao, X., Xiao, L.: Simple probabilistic algorithm for detecting community structure. Phys. Rev. E **79**, 036111 (Mar 2009). https://doi.org/10.1103/PhysRevE.79.036111
56. Richardson, T., Mucha, P.J., Porter, M.A.: Spectral tripartitioning of networks. Physical Review E **80**(3), 036111 (Sep 2009). https://doi.org/10.1103/PhysRevE.80.036111
57. Riedy, J., Bader, D.A., Jiang, K., Pande, P., Sharma, R.: Detecting communities from given seeds in social networks. Tech. rep., Georgia Institute of Technology (2011)
58. Rossetti, G., Cazabet, R.: Community discovery in dynamic networks: a survey. ACM Computing Surveys (CSUR) **51**(2), 35 (2018)
59. Rosvall, M., Bergstrom, C.T.: An information-theoretic framework for resolving community structure in complex networks. Proceedings of the National Academy of Sciences **104**(18), 7327–7331 (2007). https://doi.org/10.1073/pnas.0611034104
60. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences **105**(4), 1118–1123 (2008). https://doi.org/10.1073/pnas.0706851105

61. Sabidussi, G.: The centrality index of a graph. Psychometrika **31**, 581–603 (1966). https://doi.org/10.1007/BF02289527
62. Soundarajan, S., Hopcroft, J.: Using community information to improve the precision of link prediction methods. WWW '12 Companion Proceedings of the 21st International Conference on World Wide Web pp. 607–608 (2012). https://doi.org/10.1145/2187980.2188150
63. Valverde-Rebaza, J., Lopes, A.: Structural link prediction using community information on twitter. Proceedings of the 2012 4th International Conference on Computational Aspects of Social Networks, CASoN 2012 (2012). https://doi.org/10.1109/CASoN.2012.6412391
64. Wei, D., Zhang, X., Mahadevan, S.: Measuring the vulnerability of community structure in complex networks. Reliability Engineering and System Safety **174**, 41–52 (2018). https://doi.org/https://doi.org/10.1016/j.ress.2018.02.001
65. Xie, J., Szymanski, B.K.: Community detection using a neighborhood strength driven label propagation algorithm. In: proceedings of the 2011 ieee network science workshop. pp. 188–195. nsw '11, ieee computer society, washington, dc, usa (2011). https://doi.org/10.1109/nsw.2011.6004645
66. Xie, J., Szymanski, B.K.: Towards linear time overlapping community detection in social networks. In: Proceedings of the 16th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume Part II. pp. 25–36. PAKDD'12, Springer-Verlag, Berlin, Heidelberg (2012), `http://dx.doi.org/10.1007/978-3-642-30220-6_3`
67. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. CoRR **abs/1205.6233** (2012), `http://arxiv.org/abs/1205.6233`
68. Yang, J., Leskovec, J.: Overlapping community detection at scale: A nonnegative matrix factorization approach. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining. pp. 587–596. WSDM '13, ACM, New York, NY, USA (2013). https://doi.org/10.1145/2433396.2433471
69. Zachary, W.: An information flow model for conflict and fission in small groups. J. of Anthropological Research **33**, 452–473 (1977)
70. Zachary, W.W.: Zachary karate club network dataset – KONECT (apr 2017), `http://konect.uni-koblenz.de/networks/ucidata-zachary`
71. Zarei, M., Izadi, D., Samani, K.A.: Detecting overlapping community structure of networks based on vertex–vertex correlations. Journal of Statistical Mechanics: Theory and Experiment **2009**(11), P11013 (nov 2009). https://doi.org/10.1088/1742-5468/2009/11/p11013
72. Zhang, S., Wang, R.S., Zhang, X.: Identification of overlapping community structure in complex networks using fuzzy c-means clustering. Physica A: Statistical Mechanics and its Applications **374**, 483–490 (01 2007). https://doi.org/10.1016/j.physa.2006.07.023