Understanding Vulnerability of Communities in Complex Networks

VIRAJ PARIMI, Indraprastha Institute of Information Technology, India
ARINDAM PAL, TCS Research & Innovation, India
SUSHMITA RUJ, Indian Statistical Institute, India
PONNURANGAM KUMARAGURU, Indraprastha Institute of Information Technology, India
TANMOY CHAKRABORTY, Indraprastha Institute of Information Technology, India

In this paper, we study the crucial elements of complex networks, namely nodes, and edges and their properties such as their community structure, which play an important role in dictating the robustness of the network towards structural perturbations. Specifically, we want 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 extremely important because the community structure of a network allows deep underlying insights into how function of a network and its topology affect each other. Moreover, it even provides a way to condense large graphs into smaller graphs where each community acts as a meta node and hence aids in easier network analysis. If this community structure was to be compromised by either accidental or intentional perturbations to the network, that would make such analysis difficult. Since the problem of identifying such vital nodes is computationally intractable, we propose some heuristics that allow us to find solutions close to the optimal solution. To certify the effectiveness of our approach, we first test these heuristics on small networks, and then move to larger networks to show that we achieve similar results. The results reveal that the proposed approaches are effective to analyze the vulnerability of communities in graphs irrespective of their size and scale. From the application point of view, we show that the proposed algorithm is scalable and can be applied to the information diffusion task to curtail the spread of active nodes which was observed empirically. Additionally, we show the performance of our algorithm through an extrinsic evaluation—we employ two tasks, like prediction and information diffusion, and show that the effect of our algorithm on these tasks is higher than the other baselines.

Additional Key Words and Phrases: Community Structure, Vulnerability Assessment, Complex Networks

ACM Reference Format:
Viraj Parimi, Arindam Pal, Sushmita Ruj, Ponnurangam Kumaraguru, and Tanmoy Chakraborty. 2019. Understanding Vulnerability of Communities in Complex Networks. 1, 1 (June 2019), 20 pages. https://doi.org/10.1145/nmnmnn.mnmn

1 Introduction

A large body of research in complex networks involves study and effects of community structure as it is one of the salient structural characteristics of many real-world networks. A network is said to have a community structure if it can be grouped easily into a set of nodes such that each set of nodes is densely connected, while the interconnections between these sets are sparse. 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 first issue as described by an extensive survey conducted by Lancichinetti and Fortunato
[20]. 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 as it provides insight into
how a network functions and how does the change in the network topology affect the interactions between the
nodes within the network. Exploring this structural vulnerability of the network allows us to prepare beforehand
in case the network is affected by undesired perturbations and adversarial attacks. A major factor in order to
understand this is to analyze the network and comprehend the effect of the failure of these vital nodes 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 community structure of the network. 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 this community structure is quantified using different measures such as Modularity [23],
Normalized Mutual Information [11] and Adjusted Rand Index [19], etc.

There are many real world applications of this problem. Consider a power grid network where power outage is
a frequent occurring event. In such a scenario the vendor of this power needs to make quick decisions as to how
the failure of some nodes in the network would affect his customers. The solution would be to ensure that crucial
nodes in this network have enough backup so that restoration process can move effortlessly. Another application
would be the railway networks where inadvertent cutting of routes to certain stations can cause significant
problem to residents of the city. Hence, the government needs to ensure that routes to certain critical stations
have redundancies so that if one route gets cut off then other routes can be made use of. This problem also has
applications in Online Social Networks such as the worm containment problem [21]. This knowledge would
provide helpful insights into the protection of sensitive nodes once worms spread out into the network. In all of
the problems 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 which can be as small as removal of a node in the
network can possibly lead to the breakdown of the community that the node was a part of given that the removed
node had large influence on the network. It can also happen that if the removed node was of less significance, then
that would have very less impact on the community structure of the network. Additionally, understanding the
network vulnerability from the community structure standpoint is important in real-world setting because the
networks that are dealt with here, have huge size which add to the computational overhead and most importantly
because they shed light onto some latent characteristics shared by different nodes. Since communities can act as
meta-nodes, they allow for easier study of large networks thereby reducing the computational overhead and
provide useful insights based on the properties shared by the nodes of a community which can be exploited to
understand the vulnerability of the network as a whole entity.

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

- We study the structural vulnerability of communities in networks and assess the impact of the failure of
  nodes on the underlying community structure.
- We suggest few heuristics including a hierarchical greedy based approach which allows for identifying
  such critical nodes in the network that have profound impact on 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 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 work involves around two broad strategies, namely community structure analysis and structural network vulnerability assessment. Numerous approaches have been developed and applied to detect community structure. For instance, a hierarchical agglomerative algorithm was proposed by Newman et al. [16]. A much more extensive literature can be found in an excellent survey which describes all approaches derived to solve the problem [20].

Assessing the structural network vulnerability has received increasing attention. For example, Nguyen et al. [25] 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. [24] explored the number of connected triplets in a network as they capture the strong connection of communities in social networks.

Additionally, different measures and metrics have been proposed to measure the robustness of a network. Such measures include average size of a cluster, relative size of the largest components, diameter and graph connectivity. One approach dealt with this problem using the weighted count of loops in a network. Chan et al. [10] addressed this problem in both deterministic and probabilistic settings where they suggested solutions based on minimum node cutset. Frank et al. [14] outlined a solution that uses the second smallest eigenvalue of a laplacian matrix of a graph and termed it as the algebraic connectivity of that graph. Fiedler [13] proposed four basic attack strategies, namely, ID removal, IB removal, RD removal and RB removal. ID and RD removal deal with degree distribution of the graph where the only difference is that the second approach changes the removal strategy based on the change in the degree distribution itself. IB and RB removal are also similar constructs, but they are based in the betweenness distribution. Holme et al. [18] used an algorithm adapted from Google’s page-rank providing a sequence of losses that adds to the collapse of the network. Allesina et al. [2] evaluated the network characteristics like cyclomatic number and gamma index. They mentioned that such global graph theoretic based indices are themselves not sufficient to measure the vulnerability of a network, but they showcase the hierarchy of nodes in the system.

Ramirez et al. [26] proposed an approach where the resilience of the community structure is quantified by first 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. [17] provide a review of various approaches that make use of the facility importance concept to understand the system wide vulnerability. Sankaran et al. [30] proposed a new vulnerability metric where they considered a combination of external and internal factors such as connection density etc. These methods allow us to quantify the vulnerability of a community, but does not provide us with a set of nodes that contribute to the vulnerability of the community itself.

The information of critical nodes which contribute to the vulnerability of the communities 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 \( A \) be any community detection algorithm. For a vertex set \( S \in V \), let \( G[S] \) be the subgraph induced by \( S \) and \( f(A(G[V]), A(G[V \setminus S])) \) is a value function that computes some measure of the difference between the community structures of \( G[V] \) and \( G[V \setminus S] \) provided by \( A \). We need to identify a set \( S \in V \) of size \( k \) which,

\[
\text{maximize} \quad f(A(G[V]), A(G[V \setminus S]))
\]

This problem is computationally intractable as shown by Alim et al. [25] and hence requires the use of greedy heuristics to approach the optimal answer.

4 Preliminaries

We used Louvain algorithm for detecting the underlying community structure. It is a greedy optimization algorithm proposed by Blondel et al. [5], that tries to optimize the modularity metric of a graph and extracts communities from large graphs 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 that was designed 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,

\[
Q = \frac{1}{2m} \sum_{i,j} \left( A_{ij} \delta(c_i, c_j) - \frac{k_i k_j}{2m} \right)
\]

where \( m = \) number of edges, \( A = \) Adjacency matrix, \( k_i, k_j = \) degree of node \( v \) and node \( w \), \( c_i, c_j = \) community label of node \( v \) and node \( w \), \( \delta(c_i, c_j) = 1 \) 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.

\[
N = \frac{\sum_{i \in X} \sum_{j \in Y} n_{ij} \log \left( \frac{n_{ij}}{n_i n_j} \right) \frac{c_X}{n-k} H(Y) + \frac{c_Y}{n-k} \log(n-k) - \sum_{j \in Y} y_j \log(y_j)}}{(n-k)H(X) + \frac{c_Y}{n-k} \log(n-k) - \sum_{j \in Y} y_j \log(y_j)},
\]

where \( c_X = \) Number of communities in network \( X \), \( c_Y = \) Number of communities in network \( Y \), \( n_{ij} = |X_i \cap Y_j| \), \( n = \) Number of nodes, \( x_i = |X_i| \), \( y_i = |Y_i| \), \( g = \) 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,

\[
R = \frac{\sum_{i,j} \binom{n_{ij}}{2} - \binom{\sum_i a_i}{2} \binom{\sum_j b_j}{2}}{\binom{n}{2} \left( \binom{\sum_i a_i}{2} + \binom{\sum_j b_j}{2} \right) - \binom{\sum_i a_i}{2} \sum_j \binom{b_j}{2} + \binom{\sum_j b_j}{2} \sum_i \binom{a_i}{2}}
\]

where \( L = \) Contingency Table, \( n_{ij} = L[i][j] \), \( a_i = \) Sum of \( i \) row in \( L \), \( b_j = \) Sum of \( j \) column in \( L \), \( n = \) Number of nodes in the network

, Vol. 1, No. 1, Article . Publication date: June 2019.
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 with, thereby allowing us to quantify the effectiveness of these heuristics.

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

**Algorithm 1:** Exhaustive Algorithm

- **Input:** Network $G = (V, E)$, $k$, a community detection algorithm $A$, a value function $F$
- **Output:** Set of nodes whose size is $k$

1. $X \leftarrow$ Run community detection algorithm $A$ on $G$.
2. $C \leftarrow$ Generate all the combination of nodes in $V$ of size $k$.
3. $\text{foreach } C_i \in C$ do
4.     $G' \leftarrow$ Remove $C_i$ from $G$.
5.     $Y \leftarrow$ Run community detection algorithm $A$ on $G'$.
6.     Compute $F$ by comparing $Y$ and $X$.
7.     Return a $C_i$ which maximizes $F$.
8. end

Next we propose a naive network based greedy approach as defined in Algorithm 2. This algorithm takes in a property as an input and ranks the nodes in the input graph based on the property specified. Then, it greedily removes the top $k$ nodes based on their ranks and then evaluates the underlying community structure using a community detection algorithm. Based on the output of this algorithm it 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 used 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.
- **Degree Centrality** - It is defined as the number of edges that are incident upon a node.
- **Betweenness Centrality** - We use the betweenness centrality estimate defined by Freeman [15] as the number of times a node acts as a bridge along a shortest path route between two other nodes.
- **Eigenvector Centrality** - It is a measure of the influence of a particular node in the graph. This centrality estimate is based on the intuition that a node is more central when there are more connections within it’s local network and was introduced by Bonacich [6].
- **Closeness Centrality** - It measures how easily other vertices can be reached from a particular vertex [4, 27].
- **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 a graph is a maximal subgraph in which each vertex has at least degree $k$. It was defined by Batagelj and Zaversnik [3].
- **Diversity** - The diversity index of a vertex is estimated by the normalized Shannon entropy of the weights of the edges incident on a vertex [12].
- **Eccentricity** - It is defined as the shortest maximum distance from the vertex to all the other vertices in a graph.
- **Constraint** - Introduced by Burt [8], this measure estimates the time and energy that is concentrated on a single cluster. This measure would be higher for a node which belongs to a small network and also all the contacts are highly connected.

- **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 [7].

**Algorithm 2: Network Based Greedy Approach**

**Input**: Network $G = (V, E)$, $k$, a structural property to rank the nodes $P$, a community detection algorithm $A$, a value function $F$

**Output**: Set of nodes whose size is $k$, score

1. $X \leftarrow$ Run community detection algorithm $A$ on $G$.
2. $R \leftarrow$ Rank the nodes in $G$ based on the structural property $P$.
3. $G' \leftarrow$ Remove top $k$ nodes from $G$ based on $R$.
4. $Y \leftarrow$ Run community detection algorithm $A$ on $G'$.
5. Compute the value function $F$ by comparing $X$ and $Y$.
6. Return the set of top $k$ nodes in $R$ along with the score of the value function $F$.

The downside of this algorithm is that it does not consider the effect of the nodes on the community structure of the graphs itself. This is addressed in our next Algorithm 3 which also takes into account the underlying community structure. Here, we propose a hierarchical approach where in the first phase we choose a community based on some community centric metric and then in the second phase we select a node greedily based on its node centric properties. The community centric properties which were used are the following,

- **Link density** - $D(G) = \frac{2E}{V(V-1)}$ where $E$ is the number of edges in the graph and $V$ is the number of vertices in the graph.

- **Conductance** - $\phi(G) = \frac{s}{v}$ 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 $G$.

- **Compactness** - $C(G)$ is defined as the average shortest path lengths within the graph $G$.

The algorithms proposed above are sufficient for smaller graphs as we can evaluate them with the help of Algorithm 1 but real world networks exhibit much larger and complex systems. 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 a extrinsic task, based on the graph structure is say, $\phi$, 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 graph structure, thereby validating the selection of nodes.

Specifically, suppose that a user wants to select vulnerable nodes in a large graph such that the resultant value function score is maximized. In order 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 since the graph is large, 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 performance drop by the graph on the tasks. Since we are using the same algorithms that were used with small graphs, it is evident that the actual problem at hand of maximizing the value function is still of prime focus and only the way to validate those same results has been changed for larger graphs.

In Algorithm 4, we consider two different tasks which are described as follows to quantify the performance of the graph,
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 A, a value function F

**Output**: Set of nodes whose size if k, score

1. Function best_community(graph G, community structure X):
   1. foreach $X_i \in X$ do
      2. $G' \leftarrow$ Create a subgraph from G with only the vertices from $X_i$.
      3. $R_g \leftarrow$ Rank each such $G'$ based on the community centric property $P_c$.
      4. Return $X_i$ whose induced subgraph $G'$ ranked above others based on $R_g$.
   end

2. Function best_node(graph G, community structure X):
   1. $G' \leftarrow$ Create the subgraph $G'$ from G which is induced from X.
   2. $R_n \leftarrow$ Rank the nodes in $G'$ based on the node centric property $P_n$.
   3. Return the top node that ranks above others based on $R_n$.
   4. $X \leftarrow$ Run community detection algorithm A on G.
   5. $Y \leftarrow$ Run community detection algorithm A on G.
   while k nodes are not selected do
      6. $X' \leftarrow$ best_community(G, Y).
      7. node $\leftarrow$ best_node(G, $X'$).
      8. $G' \leftarrow$ Remove node from G and add this node to the output set.
      9. $Y \leftarrow$ Run community detection algorithm A on $G'$.
   end
   10. Compute the value function F by comparing Y and X.
   11. Return the output set of nodes and the score evaluated by the value function F.

(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 graph. Hence, the problem asks to what extent can the evolution of a complex network 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 graph such as Within-Inter-Cluster defined by Rebaza et al. [29], Modified Common Neighbors and Modified Resource Allocation defined by Soundarajan and Hopcroft [28].

(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 to explore 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 which connect separate communities. We keep $p_i \geq p_o$ as information is more likely to spread among nodes within the same community as observed by Lin et al.[22].

6 Datasets
To run our experiments extensively, we select six real-world networks with diverse size. The datasets used are follows:
Algorithm 4: Task Based Approach

**Input**: Network $G = (V, E)$, $k$, a task $T$, a community detection algorithm $A$, a value function $F$

**Output**: Set of nodes whose size if $k$, score

1. **Function** compute_task_performance($task\ T, graph\ G_1, graph\ G_2, community\ structure\ of\ G_1\ X, community\ structure\ of\ G_2\ Y$):
   2.   if $T$ is link prediction then
   3.     Create a test and train edge list based on the edge set of $G_1$.
   4.     $G'_1 \leftarrow$ Create a subgraph induced by the training set.
   5.     Apply the link prediction task using $X$ to decide on the edge probabilities on $G'_1$.
   6.     Compute the F1 score for the predicted edges.
   7.     Repeat the same process for graph $G_2$.
   8.     Compare the F1 scores for both the graphs.
   9.   else
   10.    Select a random set of seed nodes which are active by default.
   11.    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.
   12.    Repeat the process with $G_2$.
   13.    Compare the number of active nodes at the end for both $G_1$ and $G_2$.
   14.  end
   15. $X \leftarrow$ Run community detection algorithm $A$ on $G$.
   16. $S \leftarrow$ Output from Algorithm 2 or Algorithm 3 which return the target set of nodes.
   17. $G' \leftarrow$ Remove nodes in $S$ from $G$.
   18. $Y \leftarrow$ Run community detection algorithm $A$ on $G'$. 
   19. $score \leftarrow$ compute_task_performance($T, G, G', X, Y$)

1. **Karate Club** - The data was collected from the members of a karate club [1, 32]. Each node represents a member of the club 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** - This network was collected by Girvan and Newman [16], which contains the network of American football games between division IA colleges during 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 [9], 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. [9]. This dataset comprises of nodes that represent an author and an undirected edge between two authors is drawn 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 [31]. The nodes represent products and an undirected edge between two nodes represents that these products are co-purchased frequently. 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 [31]. Therefore each node is an 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.

| 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     |

Table 1. Real world networks used in our experiments.

7 Experiments

We divide this section into three subsections to cover all the value functions that were discussed in Section 4. We first present the results of Algorithm 1 for smaller graphs 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 larger graphs.

7.1 Modularity

7.1.1 **Exhaustive approach** Table 2 shows the results of the Algorithm 1 on three small scale datasets when using the modularity as the target value function. We performed the analysis by fixing $k = 5$. For the karate network, we observe that nodes $(0, 1, 3, 5, 6)$ are the most vulnerable as their removal maximized the difference of modularity scores between the original and the perturbed graph. Similarly, the most vulnerable nodes identified for other two datasets have been tabulated in Table 2.

| 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    |

Table 2. Modularity scores and the nodes selected for smaller graphs.
7.1.2 Network based greedy approach

In this section we present the results for the analysis on all the datasets of Algorithm 2. We were able to perform this analysis on all the datasets irrespective of their scale as the algorithm applied was greedy in nature and does not need much time to execute. Moreover, we fix $k = 5$ for smaller graphs as mentioned previously, but for larger graphs, such removal strategy won’t showcase significant effects. This is because removing just 5 nodes in larger graphs won’t affect the underlying community structure enough to cause major structural perturbations. So, to handle such cases we rather remove till 5% of the total nodes. Based on the Figure 1 we infer that clustering coefficient as a network based greedy metric performs better in comparison to other greedy metrics when we remove the target 5 nodes. Moreover, when we compare the maximum values attained in the smaller graphs, we see that this algorithm was not able to attain the optimal answer indicated by Table 2. For example, in the Karate network, the maximum score obtained by Algorithm 2 was around 0.05 whereas the optimal answer was about 0.13. This indicates that there is a lot of scope for improvement.

7.1.3 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 graphs and 5% for larger graphs. We compare
Understanding Vulnerability of Communities in Complex Networks

Fig. 2. Results of Community based approach over several datasets with modularity being the target value function.

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 in nature, it also is computationally efficient. Based on Table 3 we observe that Link Density performed better in comparison to the other community centric properties as the scores over all the datasets was maximum. Now that we have established that the best community centric property in a modularity difference maximization setting is link density, we present the results of the node centric properties in Figure 2. Over all the datasets we ran experiments on, we found that eigenvector centrality performs better compared to other greedy metrics. Additionally, when we compare the results of this algorithm with the ground truth data presented in Table 2 we observe that this solution comes close to the optimal solution. For example, in the Railway network, we observe that the best modularity score that is obtained to be around 0.06 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.
7.2 Normalized Mutual Information

7.2.1 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 NMI. Note that we would want to minimize NMI as the this metric gives a value of 1 for two community structures that are similar 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 5 tuple nodes will further decrease the NMI score between the two partitions. Similarly the ground truth values for the other small datasets can be found in the Table 4.

| 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|

Table 4. NMI scores and the nodes selected for smaller graphs.

7.2.2 Network based greedy approach In this section we present the results for the analysis on all the datasets of Algorithm 2. Moreover, we fix k = 5 for smaller graphs as mentioned previously, but for larger graphs, such removal strategy won’t showcase significant effects and hence we remove till 5% of the total nodes in such cases. Based on the Figure 3 we infer that eccentricity as a network based greedy metric performs better in comparison to other greedy metrics when we remove the target 5 nodes. As we are evaluating 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 the value of NMI is small when two clusterings are not same as mentioned previously in the Section 4. Based on this comparison for smaller graphs, we see that this algorithm was not able to attain the optimal answer indicated by Table 4. For example, in the Karate network, the minimum score obtained by Algorithm 2 was around 0.55 whereas the optimal answer was about 0.36. This indicates that there is a lot of scope for improvement.

7.2.3 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 graphs and 5% for larger graphs. 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 in comparison to the other community centric properties as the scores over all the datasets were minimum. With link density as the best community centric method, we present the results of the node centric properties in Figure 4. Over all the

| 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     |

Table 3. Best modularity scores using different community centric properties.
datasets we ran experiments on, we found that clustering coefficient performs better compared to other greedy metrics. Additionally, when we compare the results of this algorithm 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 observe that the best NMI score that is 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.

7.3 Adjusted Rand Index

7.3.1 Exhaustive approach Table 6 shows the results of the Algorithm 1 on three small scale datasets when using the ARI as the target value function. We performed the analysis by fixing $k = 5$. For the football network, we observe that nodes (61, 85, 16, 99, 7) are the most vulnerable as their removal minimized the ARI scores between...
Fig. 4. Results of Community based approach over several datasets with NMI being the target value function.

| 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     |

Table 5. Best NMI scores using different community centric properties.

the original and the perturbed graph’s vertex clusterings. Similarly, the most vulnerable nodes identified for other two datasets have been tabulated in Table 6.
7.3.2 Network based greedy approach In this section we present the results for the analysis on all the datasets of Algorithm 2. We fix $k = 5$ for smaller graphs as mentioned previously, but for larger graphs we remove till 5% of the total nodes. Based on the Figure 5 we infer that closeness vitality as a network based greedy metric performs better in comparison to other greedy metrics when we remove the target 5 nodes. As we are evaluating 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 the value of ARI is small when two clusterings do not agree with each other as mentioned previously in the Section 4. Based on this comparison for smaller graphs, we see that this algorithm was not able to attain the optimal answer indicated by Table 6. For example, in the Railway network, the minimum score obtained by Algorithm 2 was around 0.65 whereas the optimal answer was about -0.28. This indicates that there is a lot of scope for improvement.

7.3.3 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 graphs and 5% for larger graphs. We compare the different community centric properties in Table 7. Here we present the best ARI scores obtained after applying this algorithm on all the datasets. Based on Table 7 we observe that Conductance performed better in comparison to the other community centric properties as the scores over all the datasets were minimum. With conductance as the best community centric method, we present the results of the node centric properties in Figure 6. Over all the datasets we ran experiments on, we found that coreness performs better compared to other greedy metrics. Additionally, when we compare the results of this algorithm 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 observe that the best ARI score that is 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 2.

![Table 6. ARI scores and the nodes selected for smaller graphs](image)

| 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|

Table 6. ARI scores and the nodes selected for smaller graphs.

![Table 7. Best ARI scores using different community centric properties.](image)

| 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     |

Table 7. Best ARI scores using different community centric properties.

7.4 Task based approach

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

7.4.1 Link Prediction We tested this task by assigning probabilities to the edges using three metrics separately namely, Within-Inter Cluster, Modified Common Neighbors and Modified Resource Allocation. We found that Within-Inter Cluster gave better results compared to the other alternatives.

Based on the Figure 7, we observe that over all value functions the performance of the graph 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 reason for the performance drop can be attributed to significant changes introduced into the system by removal of vulnerable nodes. Their removal triggered major structural perturbations in the underlying community structure which caused the within-inter cluster method to assign lower probabilities to the edges due to less connections within the community and more connections
across other communities. This decreased the likelihood of test edge being classified as a valid link thereby decreasing the performance.

7.4.2 Information Diffusion Based on the Figure 8, we observe that over all value functions the performance of the graph in the information diffusion task has decreased which is evident from the lower fraction of nodes that were active. For this set of experiments, we set $p_i \geq p_o$ and let the cascade model run for 200 iterations. With 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 to that the fraction of nodes that will be active at the end of all the iterations would be low if the underlying community structure was significantly perturbed and the graph was highly disconnected whereas the 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 got from Algorithm 3 when applied to the larger graphs using Algorithm 4 resulted in the decrease in the performance of
the graphs over both tasks that they were employed on thereby validating our initial hypothesis. This establishes that Algorithm 3 can be applied to any general graph irrespective of their size.

8 Conclusion

Due to the extensive size of our experiments, we show our best results only. Since Algorithm 1 is exhaustive in nature and hence was applied only to small graphs 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 the 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 graphs. As mentioned previously we used Algorithm 4 to compare the performance of Algorithm 2 and Algorithm 3 for large graphs such as Co-authorship, Amazon and Live Journal Networks. Based on these results, we observe that when we use the results of Algorithm 3 we get a performance drop over both the tasks namely link prediction and information diffusion compared to the original graph. This establishes the generalizability of Algorithm 3.
To conclude, this work has provided a better hierarchical approach that allowed for better approximate identification of the vulnerable nodes in a network. The proposed method was used to analyze the community vulnerability of several graphs whose validity was established using both exhaustive and task based approaches depending on the size of the network.

References

[1] 2017. Zachary karate club network dataset – KONECT. http://konect.uni-koblenz.de/networks/ucidata-zachary
[2] Stefano Allesina and Mercedes Pascal. 2009. Googling Food Webs: Can an Eigenvector Measure Species’ Importance for Coextinctions? PLOS Computational Biology 5, 9 (09 2009), 1–6. https://doi.org/10.1371/journal.pcbi.1000494
[3] Vladimir Batagelj and Matjaz Zaversnik. 2003. An O(m) Algorithm for Cores Decomposition of Networks. CoRR cs.DS/0310049 (2003). http://arxiv.org/abs/cs.DS/0310049
[4] A. Bavelas. 1950. Communication Patterns in Task-Oriented Groups. Acousical Society of America Journal 22 (1950), 725. https://doi.org/10.1112/1.1906679
[5] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008, 10 (2008), P10008. http://stacks.iop.org/1742-5468/2008/i=10/a=P10008
[6] Phillip Bonacich. 1972. Factoring and weighting approaches to status scores and clique identification. The Journal of Mathematical Sociology 2, 1 (1972), 113–120. https://doi.org/10.1080/0022250X.1972.9989806 arXiv:https://doi.org/10.1080/0022250X.1972.9989806
[7] U. Brandes, T. Erlebach, Gesellschaft für Informatik, and Gesellschaft für Informatik. 2005. Network Analysis: Methodological Foundations. Springer. https://books.google.co.in/books?id=TTNhSm7HYvC
[8] RonaldÁ‡. Burt. 2004. Structural Holes and Good Ideas. Amer. J. Sociology 110, 2 (2004), 349–399. http://www.jstor.org/stable/10.1086/421787
[9] Tanmoy Chakraborty, Sriram Srinivasan, Niloy Ganguly, Animesh Mukherjee, and Sanjukta Bhowmick. 2014. On the Permanence of Vertices in Network Communities. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’14). ACM, New York, NY, USA, 1396–1405. https://doi.org/10.1145/2623330.2623707
[10] Hau Chan, Leman Akoegl, and Hanghang Tong. [n.d.]. Make It or Break It: Manipulating Robustness in Large Networks. 325–333. https://doi.org/10.1137/1.9781611973440.37
[11] Leon Danon, Albert DÁ‡az-Guilera, Jordi Duch, and Alex Arenas. 2005. Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment 2005, 09 (2005), P09008. http://stacks.iop.org/1742-5468/2005/i=9/a=P09008
[12] Nathan Eagle, Michael Macy, and Rob Claxton. 2010. Network diversity and economic development. Science 328 5981 (2010), 1029–31. https://doi.org/10.1126/science.1185726
[13] Miroslav Fiedler. 1973. Algebraic connectivity of graphs. Czechoslovak Mathematical Journal 23, 2 (1973), 298–305. http://eudml.org/doc/12723
[14] H. Frank and I. Frisch. 1970. Analysis and Design of Survivable Networks. IEEE Transactions on Communication Technology 18, 5 (October 1970), 501–519. https://doi.org/10.1109/TCOM.1970.1090419
[15] Linton C. Freeman. 1977. A Set of Measures of Centrality Based on Betweenness. Sociometry 40, 1 (1977), 35–41. http://www.jstor.org/stable/3033543
[16] M. Girvan and M. E. J. Newman. 2002. Community structure in social and biological networks. Proceedings of the National Academy of Sciences 99, 12 (2002), 7821–7826. https://doi.org/10.1073/pnas.122653799 arXiv:http://www.pnas.org/content/99/12/7821.full.pdf
[17] Tony H. Grubesic, Timothy C. Matisziw, Alan T. Murray, and Diane Snediker. 2008. Comparative Approaches for Assessing Network Vulnerability. International Regional Science Review 31, 1 (2008), 88–112. https://doi.org/10.1177/0160017607308679 arXiv:https://doi.org/10.1177/0160017607308679
[18] Petter Holme, Beom Jun Kim, Chang No Yoon, and Seung Kee Han. 2002. Attack vulnerability of complex networks. Phys. Rev. E 65 (May 2002), 056109. Issue 5. https://doi.org/10.1103/PhysRevE.65.056109
[19] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. Journal of Classification 2 (1985), 193–218. Issue 1. https://doi.org/10.1007/BF01908075
[20] A. Lancichinetti and S. Fortunato. 2009. Community detection algorithms: A comparative analysis. Physical Review 80, 5 (2009). https://doi.org/10.1103/PhysRevE.80.056117
[21] P. Li, M. Salour, and X. Su. 2008. A survey of internet worm detection and containment. IEEE Communications Surveys Tutorials 10, 1 (First 2008), 20–35. https://doi.org/10.1109/COMST.2008.4483668
[22] Shuyang Lin, Qingbo Hu, Guan Wang, and Philip S. Yu. 2015. Understanding Community Effects on Information Diffusion. In Advances in Knowledge Discovery and Data Mining, Tzu Cao, Ee-Peng Lim, Zhi-Hua Zhou, Tu-Bao Ho, David Cheung, and Hiroshi Motoda (Eds.). Springer International Publishing, Cham, 82–95.
[23] M. E. J. Newman. 2006. Modularity and community structure in networks. Proceedings of the National Academy of Sciences 103, 23 (2006), 8577–8582. https://doi.org/10.1073/pnas.0601602103 arXiv:http://www.pnas.org/content/103/23/8577.full.pdf
[24] H. T. Nguyen, N. P. Nguyen, T. Vu, H. X. Hoang, and T. N. Dinh. 2017. Transitivity Demolition and the Fall of Social Networks. *IEEE Access* 5 (2017), 15913–15926. https://doi.org/10.1109/ACCESS.2017.2672666

[25] Nam P. Nguyen, Abdul Alim, Yilin Shen, and My T. Thai. 2013. 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)* (2013), 231–235. https://doi.org/10.1109/2492517.2492644

[26] Jose E. Ramirez-Marquez, Claudio M. Rocco, Kash Barker, and Jose Moronta. 2018. Quantifying the resilience of community structures in networks. *Reliability Engineering and System Safety* 169 (2018), 466–474. https://doi.org/10.1016/j.ress.2017.09.019

[27] Gert Sabidussi. 1966. The centrality index of a graph. *Psychometrika* 31 (1966), 581–603. Issue 4. https://doi.org/10.1007/BF02289527

[28] Sucheta Soundarajan and John Hopcroft. 2012. Using community information to improve the precision of link prediction methods. *WWW ‘12 Companion Proceedings of the 21st International Conference on World Wide Web* (2012), 607–608. https://doi.org/10.1145/2187980.2188150

[29] Jorge Valverde-Rebaza and Alneu Lopes. 2012. 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

[30] Daijun Wei, Xiaoge Zhang, and Sankaran Mahadevan. 2018. Measuring the vulnerability of community structure in complex networks. *Reliability Engineering and System Safety* 174 (2018), 41–52. https://doi.org/10.1016/j.ress.2018.02.001

[31] Jaewon Yang and Jure Leskovec. 2012. Defining and Evaluating Network Communities based on Ground-truth. *CoRR* abs/1205.6233 (2012). arXiv:1205.6233 http://arxiv.org/abs/1205.6233

[32] Wayne Zachary. 1977. An Information Flow Model for Conflict and Fission in Small Groups. *J. of Anthropological Research* 33 (1977), 452–473.