applied sciences

MDPI

Article

Node-Based Resilience Measure Clustering with Applications to Noisy and Overlapping Communities in Complex Networks †

John Matta [1], Tayo Obafemi-Ajayi [2], Jeffrey Borwey [3], Koushik Sinha [4], Donald Wunsch [5] and Gunes Ercal [1,*]

[1] Department of Computer Science, Southern Illinois University Edwardsville, Edwardsville, IL 62025, USA; jmatta@siue.edu
[2] Department of Engineering, Missouri State University, Springfield, MO 65897, USA; tayoobafemiajayi@missouristate.edu
[3] Google, Inc., Mountain View, CA 94043, USA; jeffborwey@gmail.com
[4] Department of Computer Science, Southern Illinois University Carbondale, Carbondale, IL 62901, USA; koushik.sinha@cs.siu.edu
[5] Electrical and Computer Engineering Department, Missouri S & T, Rolla, MO 65409, USA; dwunsch@mst.edu
* Correspondence: gercal@siue.edu; Tel.: +1-618-650-3348

† This paper is an extended version of our paper published in ICDM 2016.

Received: 12 July 2018; Accepted: 28 July 2018; Published: 6 August 2018

Abstract: This paper examines a schema for graph-theoretic clustering using node-based resilience measures. Node-based resilience measures optimize an objective based on a critical set of nodes whose removal causes some severity of disconnection in the network. Beyond presenting a general framework for the usage of node based resilience measures for variations of clustering problems, we experimentally validate the usefulness of such methods in accomplishing the following: (i) clustering a graph in one step without knowing the number of clusters a priori; (ii) removing noise from noisy data; and (iii) detecting overlapping communities. We demonstrate that this clustering schema can be applied successfully using a wide range of data, including both real and synthetic networks, both natively in graph form and also expressed as point sets.

Keywords: complex networks; clustering; data mining; graph theoretic algorithms

1. Introduction

One of the most interesting and widely studied properties of complex networks is community structure. While an exact definition of a *community* is difficult to find, communities or clusters are often interpreted as groups of nodes that are more connected to each other than to the rest of the network. Because of this, nodes in a community "probably share common properties and/or play similar roles" within the network [1]. Depending on the context, *community detection* may also be referred to as *clustering* or *graph partitioning*.

Clustering is a very useful data exploratory machine learning tool that allows us to make better sense of heterogeneous data by grouping data with similar attributes based on some criteria. Popular graph partitioning methods such as the Girvan–Newman algorithm [2], sparsest-cuts [3], spectral partitioning [4], and general conductance based methods (related to spectral methods via Cheeger's inequality [5]) may be viewed as solving an *edge-based resilience* problem on a graph while simultaneously outputting the components resulting from the removal of the *critical edge set* as the set of clusters. In contrast to these graph-theoretic edge-based resilience methods, in preliminary work [6], we introduced a *node*-based resilience clustering approach using *vertex attack tolerance* (VAT) [7–10] with some unique applicability for noisy datasets. A node-based resilience measure by definition must express the relative size of a most critical set of target vertices whose removal, upon an attack, would be detrimental to the remaining network and attempt to quantify the amount of resulting damage [10].

We are the first to establish a relationship between combinatorial measures of node-based resilience and graph-theoretic clustering. The current work relies on this relationship to develop a clustering framework that can be used with many different resilience measures, the choice of which may depend upon the clustering objective and domain specific properties. To examine the relevance of different resilience measures to the problem of clustering, we perform extensive tests using the framework with the resilience measures discussed in [10], including VAT, integrity [11], toughness [12], tenacity [13], and scattering number [14]. A strength of the framework is that other resilience measures can also be used.

Our focus on *node*-based resilience measures yields a number of unique advantages in the clustering context compared to *edge*-based approaches, along with some challenges as well. Measuring the resilience of a network against targeted node attacks naturally yields both a partial clustering with some outliers or noisy data points removed and a semi-clustering, targeting potential overlap nodes with multiple neighboring components. For a complete clustering in the traditional setting, an assignment of each critical attack set node to a unique cluster must additionally be made.

The preliminary results of our node-based resilience clustering methodologies (NBR–Clust) [6,15] have demonstrated their effectiveness for clustering native point-set data in the presence of noise as well as situations in which the number of clusters is not known a priori. Whereas our preliminary works examined applications of the NBR–Clust framework to point-set data, this paper exhaustively examines applications to several types of native *network* data in addition to examining the applicability of the framework towards the important problem of overlapping community detection.

In this paper, we formalize and extensively evaluate the efficacy of the NBR–Clust framework. Specifically, the key contributions of this work are as follows:

- An empirical analysis of the robustness and accuracy of the results of NBR clustering is obtained by applying it to a diverse and rich set of data, including real and synthetically generated datasets with and without noise, as well as diverse graph networks with and without overlaps.
- It is demonstrated that NBR–Clust in many cases attains complete clustering in one step.
- The performance of NBR clustering to detect and remove noise or outliers is evaluated.
- The utility of using different resilience measures within the framework for achieving different purposes is examined.
- The application of NBR clustering to datasets with overlap is investigated, including identifying overlap nodes and assigning them to multiple clusters.

The practical utility of the NBR–Clust clustering method described in this paper is further demonstrated in [16–18], where it is applied to new medical and biological datasets. In [16], the method is applied to a database of Autism Spectrum Disorder phenotypes. Results for that dataset showed that the resilience measure *tenacity* gave the best clustering results, and the minimum connectivity k-Nearest Neighbor (kNN) graph (as defined in Section 4.1.1) was the optimal representation for that data, in accordance with evidence presented in [15,19]. In [17], DNA genomic data was clustered in order to determine biological origins and genetic similarity of worldwide individuals. *Integrity* was the node-based resilience measure used due to its robustness when the number of ground truth clusters is unknown. NBR–Clust's ability to detect noise and overlaps was a useful advantage with the genomic data because individuals detected as overlaps were presumed to have a mixed genetic origin. In [18], a study of sources of resistance of grapevines to powdery mildew disease, genetic data was converted to geometric graphs and clustered. This article complements the work of [16–18] in several ways. First, this paper provides a theoretical justification for the NBR–Clust framework and demonstrates a class of graphs under which it performs better than spectral clustering and modularity. Second, we increase the scope of uses for the NBR–Clust framework by applying it to the overlapping community detection problem. Third, while the previous papers only consider point-set data transformed into graphs, here we consider a wide variety of synthetic network models in addition to well-known real network datasets from the community detection literature. Finally, in this paper, the algorithms have been generalized to include parallel approximations that allow the application of the framework to much larger network datasets.

2. Related Work

2.1. Graph Theoretic Clustering Methods

Graph-theoretic techniques for clustering are important not only when the data is expressed in a network structure, but also due to the established effectiveness of graph partitioning techniques when varying types of input data are converted to a graph representation [20]. Graph-theoretic clustering algorithms based on spectral methods are amongst the most rigorously studied from a theoretical perspective [3,4]. A well-known optimization framework for graph partitioning involves finding *sparse* cuts either to k-partition the vertices with multi-way cuts [3,4], or, more commonly, to recursively *bi-partition* the graph using the combinatorial measure *conductance* to quantify the sparsity of the cuts [21]. However, the hardness of the underlying combinatorial optimization [22] makes these methods less practical to apply without heuristics.

Commonly used methods for community detection as well as the challenges related to these methods are discussed in [1,23,24]. The Girvan–Newman algorithm is a well-known method in which a network's highest betweenness edges are greedily removed to obtain a top-down hierarchical clustering [2]. This paper's NBR–Clust framework also uses betweenness centrality calculations as a subroutine towards heuristically computing resilience measures. However, the two approaches are otherwise completely divergent as this is only a heuristic rather than an inherent aspect of the NBR–Clust framework. This study also uses *vertex* betweenness centralities rather than the *edge* betweenness centralities used in Girvan–Newman. The Girvan–Newman algorithm is not based on any global resilience computations, neither edge-based nor node-based resilience measures.

Newman [25] and many others have developed heuristics to approximate a solution to the NP-complete problem of modularity optimization, which may provide a stopping condition for Girvan–Newman community detection. These modularity-based clustering approaches start with some partitioning of the vertex set and modify the groupings in a manner that provides a maximum increase in modularity from one iteration to the next. This class of methods is similar to internal validation-based methods for traditional clustering. While such methods may be used in post-processing any existing clustering, NBR–Clust shares little in common with this class of techniques.

2.2. Noise and Outlier Detection

In the context of this work, the terms noise and outlier are used interchangeably. An outlier can be defined as "an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism" [26]. In [27], noise is defined as anything that obscures the relationship between the features of an instance and its class or observations resulting from non-systematic errors. Outliers are known to bias clustering results significantly, especially when the underlying assumption is that every data point has to reside in a cluster.

In [28], clustering and outlier detection is modeled as an integer programming optimization task which requires prior knowledge of the number of outliers. Chawla and Gionis [29] present a method for simultaneously clustering and reporting outliers that is applicable for k-means clustering. Obafemi-Ajayi et al. [30] present an iterative method based on visualization using hierarchical agglomerative methods to identify and remove outliers. Algorithmic solutions for outlier detection have also been proposed in [31,32]. With NBR–Clust, most noise is identified as part of a resilience measure's critical attack set. Additional noise can be identified using separability, a graph-based internal validation measure, as will be further explained in Section 4.1.3.

2.3. Overlapping Communities

The study of overlapping communities has become increasingly important [33], particularly due to its applicability to social networks. Extensive surveys of algorithms for community detection with overlap are presented in [1,33]. Traditional clustering methods do not necessarily work in the presence of overlaps, and in [34] it is speculated that the ineffectiveness of many clustering algorithms on large datasets is due to the fact that overlapping nodes effectively join communities together, preventing algorithms from detecting them.

In the context of overlapping communities, one of the most popular detection methods is clique percolation [35]. Clique percolation assumes that edges within a cluster are more likely to be densely connected (i.e., form cliques) than edges between clusters. The method builds overlapping communities by identifying and linking adjacent cliques. This bottom-up approach contrasts substantially with NBR–Clust, which separates a graph by removing a critical attack set to find communities.

An edge-based approach is taken with link communities [36], where communities are identified as sets of closely related links, as opposed to sets of nodes sharing many links. In this algorithm, a dendrogram is constructed from the relationships between links and is then cut at a level that obtains the desired number of clusters. The method accounts for overlapping nodes by considering that, while links have a unique position on the dendrogram, nodes can occupy multiple positions.

The RaRe (Rank Removal) algorithm of [37] is a node-centric pre-processing method where "important" nodes (e.g., characterized by high page-rank) are identified and removed until the core components reach a given size. While both RaRe and NBR–Clust are node-based methods contrasting much of the edge-based methods existing in literature, there are fundamental significant differences between the two approaches. Unlike RaRe, which requires at least two significant stages of post-processing following the removal of *individual* heuristically important nodes, the determination of the critical attack *set* of nodes via rigorously established node-based resilience measures forms the basis of accurate and effective NBR clustering. Moreover, unlike the removed nodes of RaRe, the critical attack set of nodes computed via NBR–Clust is always a super-set of the overlap nodes and a subset of the outlier nodes in noisy datasets. The motivation for NBR–Clust was extending the relationship between clustering and edge-based resilience notions to node-based resilience notions. By specifically targeting inter-cluster nodes, NBR–Clust obtains high quality components initially, requiring only light post-processing via merging the components in some situations.

3. NBR Measures in Clustering

Definitions of Node Resilience Measures

Resilience measures attempt to quantify the cost of an attack on a network in proportion to the amount of disruption the attack causes. A common measure of cost is the size of the critical attack set. The attack set is defined as a set of nodes whose removal causes disruption to a network, dividing it into disconnected components. An attack set can also be referred to as a node-cut, a cut set or a separator. The disruption created by the removal of an attack set must also be quantified. Common ways of quantifying disruption include the number of resulting components (the higher the value, the greater the disruption) or the size of the largest remaining connected component (the lower the value, the greater the disruption). For comparisons of various resilience or vulnerability measures, including those involved in this work, we refer the reader to [10]. An observation one may make about all of these NBR measures is that they return critical attack sets likely to consist of inter-cluster boundary nodes, bridge nodes and bottlenecks. The components created by removing the critical attack set are employed as the basis for candidate clusters.

Given an undirected connected graph $G = (V, E)$, the NBR measures used in this work are as follows:

VAT [7,8] is defined as

$$\tau(G) = \min_{S \subset V} \left\{ \frac{|S|}{|V - S - C_{max}(V - S)| + 1} \right\}, \tag{1}$$

where S is an attack set and $C_{max}(V - S)$ is the largest connected component in $V - S$. Vertical bars in the notation represent the cardinality of a set, such that $|S|$, for example, represents the cardinality of the set S.

Normalized integrity [11] is defined as

$$I(G) = \min_{S \subset V} \left\{ \frac{|S| + |C_{max}(V - S)|}{|V|} \right\}. \tag{2}$$

Toughness [12] is defined as

$$t(G) = \min_{S \subset V} \left\{ \frac{|S|}{\omega(V - S)} \right\}, \tag{3}$$

where $\omega(V - S)$ is the number of connected components in $V - S$.

Tenacity [13] is defined as

$$T(G) = \min_{S \subset V} \left\{ \frac{|S| + |C_{max}(V - S)|}{\omega(V - S)} \right\}. \tag{4}$$

Inverse scattering number [14] is defined as

$$h(G) = \min_{S \subset V} \left\{ \frac{1}{\omega(V - S) - |S|} \right\}. \tag{5}$$

This paper compares results using the above five resilience measures to the popular *spectral clustering* and *Louvain* methods. Spectral clustering uses combinatorial conductance, and Louvain is based on a measure called modularity. For completeness, combinatorial conductance and modularity are defined below.

Combinatorial conductance or edge based conductance [5,38] is defined as

$$\begin{aligned} \Phi(G) &= \min_{S \subset V, Vol(S) \leq Vol(V)/2} \left\{ \tfrac{|Cut(S,V-S)|}{Vol(S)} \right\} \\ &= \min_{S \subset V, Vol(S) \leq Vol(V)/2} \left\{ \tfrac{|Cut(S,V-S)|}{\delta_S |S|} \right\}, \end{aligned}$$

where $|Cut(S, V - S)|$ is the size of the cut separating S from $V - S$, $Vol(S)$ is the sum of the degrees of vertices in S, and δ_S is the *average* degree of vertices in S.

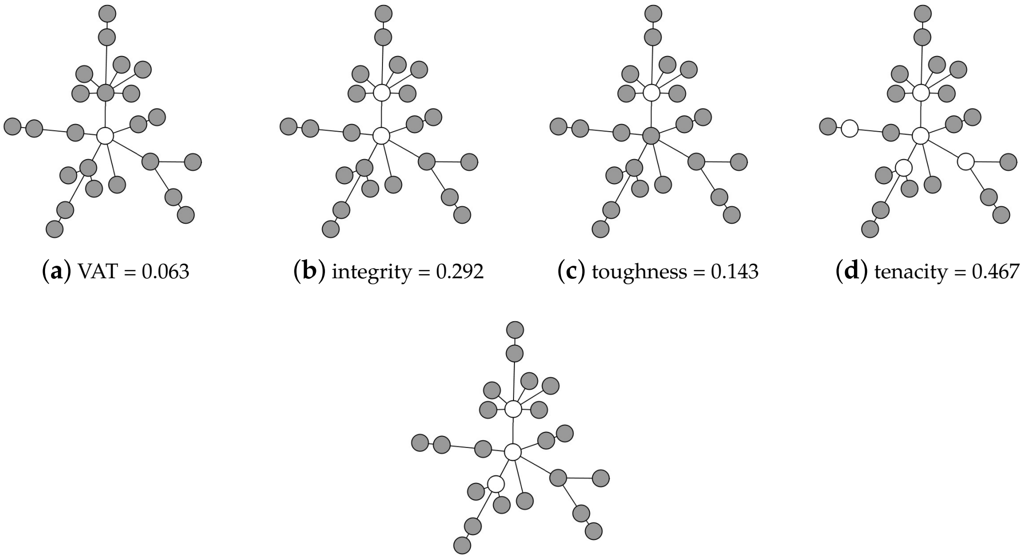
Modularity is a way of measuring the goodness of a partitioning of a graph. It is defined as

$$Modularity(G) = \sum_{P=1}^{n_P} \left[\frac{m_i}{m} - \frac{(2m_i + m_e)^2}{4m^2} \right], \tag{6}$$

where $P_1, .., P_{n_P}$ are components of a partitioning, $m = |E|$, $m_i = |\{(u, v) \in E : u \in P_i, v \in P_i\}|$, and $m_e = |\{(u, v) \in E : u \in P_i, v \notin P_i\}|$.

Other important graph measures exist which could be used within the NBR–Clust framework. Although we do not use it here, one such example is *entropy* [39], which could be used to rank the importance of nodes as in [40].

An example of the effectiveness of multiple NBR measures to cluster a simple graph is illustrated in Figure 1. For each graph, the white nodes denote the attack set returned by the corresponding resilience measure. For example, in Figure 1a, the attack set returned by VAT is the center node. Removing this node creates six disconnected clusters, the smallest of which is one node and the largest of which is eight nodes.



(**a**) VAT = 0.063 (**b**) integrity = 0.292 (**c**) toughness = 0.143 (**d**) tenacity = 0.467

(**e**) scattering = 0.091

Figure 1. Attack sets returned by the resilience measures used in this work.

In NBR clustering, the number of clusters (≥2) obtained cannot be predicted a priori. If more clusters are desired than the m clusters resulting from removing the attack set (e.g., more than $m = 6$ for Figure 1a in which VAT is applied), then one of the clusters can be further divided by calculating its VAT and removing the resulting attack set. This step can be iterated until the desired number of clusters is obtained. If fewer than $m = 6$ clusters are desired, existing clusters can be combined. Each NBR measure is based on different properties of graphs, and thus returns a different attack set, as illustrated in Figure 1. It is possible that different NBR measures will produce better results depending on the types of graphs clustered and the properties of the desired clustering, such as with or without noise.

Another advantage of node-based clustering, especially in contrast to edge-based, is that all nodes are not automatically clustered. In particular, critical attack set nodes remain unassigned after the initial candidate clusters are detected. If traditional graph partitioning is desired, each attack set node can then be assigned to a single cluster. If attack set nodes are considered to be noise (and therefore do not belong to any cluster), they can remain unassigned. If attack set nodes are likely to contain overlaps, they can be assigned to multiple clusters. As an example, the attack set for integrity, shown in Figure 1b, consists of two nodes whose removal results in 11 clusters. If a traditional clustering is desired, the two critical attack set nodes will each have to be assigned to one of the 11 clusters. One possible strategy is to assign a node to a cluster that is adjacent to it, preferably to the cluster with which it shares the most edges. The top critical node shares one edge with six different clusters, resulting in a tie that will have to be broken. On the other hand, if the node is considered to be an overlap, it can be assigned to some combination of the six adjacent clusters. These examples motivate the relevance and usefulness of NBR measures in the context of noise removal and overlap detection.

4. Theoretical Motivation of NBR–Clustering

Although the bulk of results in this work are empirical, our original motivation for considering node-based resilience measures for the clustering problem is theoretical. We first observe that, in well-known clustering algorithms, including sparsest-cuts [3], spectral clustering [4], and conductance [5], an edge-based resilience problem is solved such that the components resulting from the removal of the critical edge-cutset is output as candidate clusters. Then, we ask what would be the efficacy and special properties of a clustering algorithm founded upon *node*-based resilience measures instead.

Towards exploring this question, we must naturally examine both (i) the relationship between edge-based resilience and node-based resilience as well as (ii) the fundamental differences between them. Due to the classical importance of conductance-based clustering methods, intimately related to both sparsest-cuts and spectral clustering due to Cheeger's inequality, as well as the mathematical significance of conductance in general, we first take conductance to be a representative edge-based resilience measure. In recent work [10], VAT was noted to exhibit desirable characteristics compared to a host of other node-based resilience measures considered. As such, towards exploring item (i), we observe the following bounds proven in [8–10] relating VAT to both conductance and spectral gap for the case of regular degree graphs, hinting at some similarity in expected results between VAT-based clustering and spectral clustering for constant-degree almost-regular graphs: For any d-regular connected graph $G = (V, E)$ with λ_2 denoting the second largest eigenvalue of G's normalized adjacency matrix and $\Phi(G)$ denoting the conductance of G,

$$\frac{1}{d}\Phi(G) \leq \tau(G) \leq d^2\Phi(G). \tag{7}$$

Moreover,

$$\frac{\tau(G)^2}{2d^4} \leq 1 - \lambda_2 \leq 2d\tau(G). \tag{8}$$

While seminal graph families such as Erdos–Renyi graphs do indeed have almost-regular degree distributions in expectation, a preponderance of real data suggest highly scale-free or generally irregular degree distributions for many complex networks. On such highly variant degree distributions significant discrepancies between edge-based resilience and node-based resilience may be exhibited: As shown in [10], extremal discrepancy between node-based resilience measures and edge-based resilience measures arises in the star family of graphs, which have maximal conductance but minimal VAT. While the applicability of the star graphs in clustering problems may be suspect, two generalizations of the star family into star-of-cliques and the hypergraph $Kstar - r - uniform$ are indeed relevant for clustering. We now examine the behavior of node-based resilience measures versus edge-based resilience measures on these relevant generalizations.

As in preliminary work [15], consider the star-of-cliques graph family $K_\beta - Star(\alpha) = (V, E)$ with $n = |V| = \alpha\beta + 1$ constructed by connecting one central node v_c to each of $j = 1, \ldots, \alpha$ disjoint β-cliques K_β^j via α edges of the form $\{v_c, v_j\}$. In particular, for each clique K_β^j, there is a unique vertex v_j that is adjacent to the central vertex v_c, and we may refer to the set of α vertices adjacent to v_c as V_{adj}. An example graph with $\alpha = \beta = 5$ is shown in Figure 2. Any reasonable NBR measure for this graph, including all measures considered herein, will output either $\{v_c\}$ or V_{adj} as the critical attack set $S \subset V$ optimizing the NBR measure. As such, the removal of the critical attack set of a node-based resilience measure will result in either exactly α components corresponding identically to the α cliques K_β or it will result in exactly $\alpha + 1$ components which include the central node and α cliques $K_{\beta-1}$. In other words, the computation of a node-based resilience measure results in the immediate partitioning of the star-of-cliques such that every clique corresponds to a distinct cluster, either in its entirety or with only a single node removed.

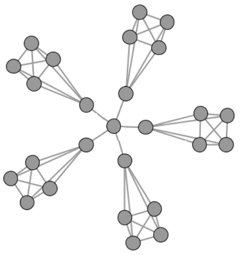


Figure 2. An extremal example illustrating the efficacy of node-based resilience measures, which attack the central node, resulting in five clusters.

On the other hand, when considering the conductance of this same star-of-cliques family, it may be observed that, for any setting of $\alpha, \beta > 1$, there are several ties for the critical attack set of edges corresponding to conductance: in fact, conductance is minimized by *any* edge-cutset H that is a subset of $E_c = \{\{v_c, v_j\} | v_j \in V_{adj}\}$ and hence cannot adequately cluster this graph. The behavior of modularity is dependent on the parameters α and β, where there exist settings of those parameters such that modularity will group some of the cliques together into a cluster. For example, if one takes $\alpha = 1024$ and $\beta = 4$, then the modularity objective function will result in fewer than 1020 clusters. Therefore, while NBR-measures yield the natural clusters for the star-of-cliques graphs regardless of α and β, edge-based resilience measures do not.

In addition, we shall now demonstrate that this is also the case for the graph representation of the $Kstar - r - uniform$ hypergraph [41] with respect to the graph representation mapping hyperedges to cliques [42]. As defined in [41], the *star hypergraph* has center node v_c and is formed by some number of hyperedges containing vertex v_c. When the number of vertices involved in every hyperedge of the star hypergraph is the same number r, then we obtain the $KStar - r - uniform$ hypergraph. For consistency of notation here, we may refer to the graph representation of the $KStar - r - uniform$ hypergraph with α hyperedges and $\beta = r$ simply as the graph $KStar(\alpha, \beta)$. Note that like $K_\beta - Star(\alpha)$, $KStar(\alpha, \beta)$ also involves a central node v_c connected to α cliques K_β. However, unlike the star-of-cliques graphs where v_c is connected to each clique K_β^j via a single connection $v_j \in V_{adj}$, in the $KStar(\alpha, \beta)$ graphs the central node v_c is connected to *all other nodes*. Thus, any vertex separator S for $KStar(\alpha, \beta)$ must include v_c and is not improved by additional nodes. Therefore, it is straightforward to see that all node-based resilience measures will output only v_c as the critical attack set for $KStar(\alpha, \beta)$. We may also observe that the conductance of $KStar(\alpha, \beta)$ suffers a similar problem as the conductance of $K_\beta - Star(\alpha)$, as there are several ties for the critical attack set of edges corresponding to conductance: Letting E_j denote all the edges connecting v_c to clique K_β^j, for any $K \subset \{1, 2, \dots \alpha\}$, conductance is minimized by edge-cutset $H = \bigwedge_{i \in K} E_i$. Hence, conductance cannot adequately cluster $KStar(\alpha, \beta)$. Finally, regarding the behavior of modularity on $KStar(\alpha, \beta)$, we again note the dependence on α and β, as well as the specific case $\beta = 4$, $\alpha = 1024$ providing a counter-example to the efficacy of modularity in detecting all of the clusters, which are naturally the cliques representing the hyperedges.

Therefore, we have exhibited classes of graphs for which NBR-measures output all of the natural clusters although some well-known clustering algorithms based on conductance or modularity may fail to do so. As no clustering algorithm is expected to work perfectly in all scenarios, this is neither a criticism of established clustering methods nor a zealous celebration of the NBR-clustering framework presented herein. Nor have we attempted to exhaustively examine graph families for which NBR–Clust has special positive clustering properties. Rather, we simply solidify our theoretical motivation for the NBR–Clustering framework.

Upon reflection of the star generalizations star-of-cliques and KStar presented above, one sees that the natural belonging of the central node in an attempted clustering is unclear. For example, as in the hypergraph $KStar(\alpha, \beta)$, the central node v_c could naturally represent the *overlap* of all the clique-clusters, or, as in the star-of-cliques, v_C might be overlap, noise, a cluster of its own, or attached to any existing clique-cluster. Although much of classical clustering research has been dedicated to situations in which a complete partitioning of the vertex set is desired, a body of recent work has been dedicated to examining the reality of incomplete partitionings due to noise or overlap. Based on our theoretical observations consistent with the examples presented herein, we hypothesize the following:

- Noise and overlap nodes will tend to be a subset of the *critical attack set* of meaningful NBR measures.
- There exist meaningful NBR measures such that a single computation of the NBR measure and removal of the corresponding critical attack set results in the natural collection component clusters.

In the remainder of this work, we present the specifications of our NBR–Clust framework across disparate scenarios and extensive empirical results supporting both of the above hypotheses. We refer to the second hypothesis as "one shot" clustering, as opposed to the repeated hierarchical applications often required of edge-based methods such as conductance. As a precursor to some of the results that follow, we note that three NBR measures have been found particularly effective in the clustering context: integrity, tenacity, and VAT. Amongst these, integrity and tenacity shall turn out particularly useful for one-shot clustering, with the number of clusters output by tenacity often presenting an upper bound on the ground truth number of clusters. Now, we present our actual framework.

4.1. NBR Clustering Framework

Given a resilience measure R (it is assumed that R expresses a minimization objective taken over possible attack sets of nodes S) and the resilience measure of a specific graph G denoted by $R(G)$, the general NBR measure R-clustering framework, NBR–Clust, is as follows:

(1) If not G, transform point data into a graph G.

(2) Approximate $R(G)$ with an acceptable accuracy, and return candidate attack set S whose removal results in a number of candidate groupings (components).

(3) Adjust the number of candidate groupings. If there are too many groups, combine them until the desired number of components is obtained. If there are too few groups, perform hierarchical clustering by proceeding recursively on the component C_i with the lowest resilience value $R(C_i)$. Continue hierarchical clustering until the desired number of groups is obtained.

(4) For a complete (i.e., traditional partition-style) clustering outcome, perform a node-assignment strategy that assigns each node v of S to the component C (from step (3)) in its original semi-partition with which v shares the most edges. Break ties arbitrarily.

(5) To cluster in the presence of noise (which implies removal of some nodes identified as noise), remove S. It is possible that some components consist entirely of noise (often significantly smaller components). In such cases, identify and remove those noise components.

(6) For clustering of networks with overlap, detect and assign overlap nodes appropriately.

The implementation of this clustering framework is available on the project website [43].

The NBR–Clust framework has great versatility in that it can be used on many different types of graphs, depending on the resilience measure R employed in step (2). Classical resilience measures such as those employed in this work are in most cases only meaningful on connected graphs. However, it is shown in [10] that the measures can be extended to account for disconnected and directed graphs. If a resilience measure can be applied to a particular graph, such as a directed, multi or binary graph, then the NBR–Clustering framework can also be applied. The following sections describe the implementation of parts of the NBR clustering framework.

4.1.1. Transforming Point Data into Graphs

In applying the NBR–Clust framework to point datasets, we first convert the data into a k-nearest neighbor (kNN) graph G. In a kNN graph G_k, vertices u and v have an edge between them if v

is amongst the k closest vertices to u with respect to the distance metric considered. While any distance metric may be used to determine nearness of neighbors, we use the n-dimensional Euclidean distance, where n is the number of features considered. Min-conn k implies choosing the minimal k such that $\forall\ k' \geq k\ \forall_{(u,v \in V)}\ \exists\ u$-$v$ path in $G_{k'}$. The choice of kNN graphs is motivated by [6,44] and empirical evidence from [15,19]. Given that NBR measures are meaningful only on connected graphs (the measure VAT has been extended in [10], such that the VAT of a disconnected graph can be considered to be the minimum VAT of any of its components), the smallest k for which the graph remains connected is chosen. An example of a synthetic point set utilized in this work and its corresponding minimum connectivity kNN graph are shown in Figures 3 and 4, respectively.

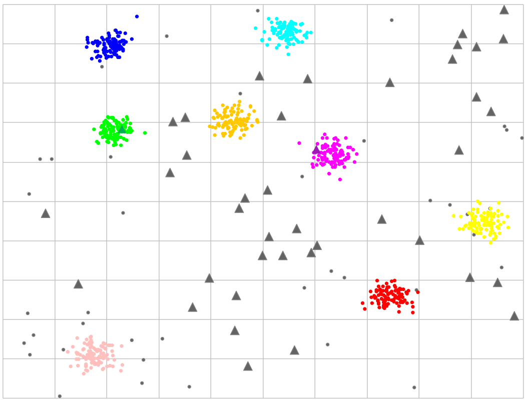


Figure 3. Visualization of a synthetic generated D2K8 dataset used in this work. Ground truth clusters are each represented by a separate color. Gray nodes are noise nodes. Gray triangles represent noise nodes detected by the NBR–Clust algorithm. Equivalent colors match corresponding clusters in Figure 4.

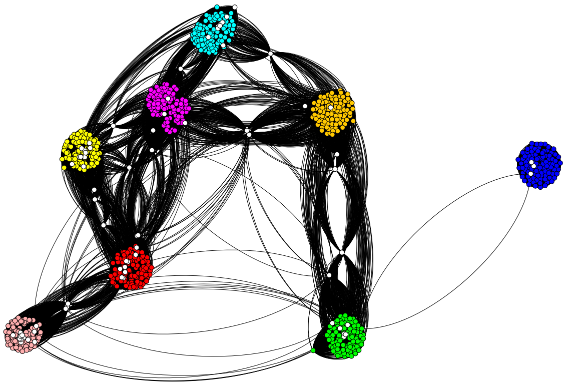


Figure 4. Visualization of the graph formed from the D2K8 dataset. Ground truth clusters are each represented by a separate color.

4.1.2. Joining Clusters

It is possible that fewer clusters are desired than the number obtained from NBR–Clust. Hence, clusters are joined based on their shared edges. For each pair of components C_A, C_B that result from the initial clustering, a normalized cut is computed and defined as follows:

$$\frac{|\{(u,v) \in E : u \in C_A, v \in C_B\}|}{|C_A||C_B|}. \tag{9}$$

Equation (9) gives a larger result for smaller components with relatively more edges between them. The two components with the largest result are joined, breaking ties arbitrarily. This creates one fewer component. This process is repeated until the desired number of components is obtained.

4.1.3. Detecting Noise Nodes in Data

When clustering in the presence of noise, the critical attack set S can be regarded as consisting entirely of noise nodes, as mentioned in step (5) of the NBR–Clust framework. We must also address the possibility of obtaining some clusters that consist of all noise nodes, which we term "all-noise" clusters in addition to S. These all-noise clusters are usually small compared to the other clusters. They are identified by using separability, a graph-based internal validation measure [45]. The separability of a set of nodes P is defined as the ratio of the number of edges in P to the number of edges on the boundary of P:

$$Separability = \frac{|\{(u,v) \in E : u \in P, v \in P\}|}{|\{(u,v) \in E : u \in P, v \notin P\}|}. \tag{10}$$

We have found empirically that all-noise clusters have below average separability values. Therefore, we use below-average separability as a threshold value to indicate all-noise clusters.

4.1.4. Identifying and Assigning Overlap Nodes

The underlying hypothesis is that overlap nodes (nodes that belong to more than one cluster) are a subset of the critical attack set nodes generated by NBR–Clust. Overlap nodes and attack set nodes share a similar characteristic: both lie on inter-cluster boundaries. In addition, an overlap node maintains a "tight connection" with all its communities [46]. An overlap node should have equally strong numbers of adjacencies to several communities, while a non-overlap node would display a significantly higher number of adjacencies to one particular community in contrast to others. An example is shown in Figure 5. Both green and cyan nodes have high betweenness centrality and are likely to be part of a resilience measure critical attack set. The cyan node is determined to be an overlap node because it is tightly connected to both the magenta and yellow communities, and those communities share no other connections. The green node is also connected to two communities but much more tightly to the gray community (four adjacencies) than to the magenta community (only one adjacency). The green node is therefore not an overlap node.

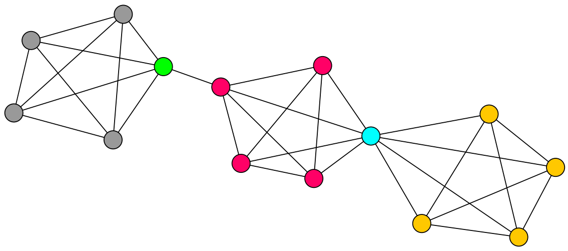


Figure 5. Identifying overlap. The cyan node overlaps the magenta and yellow clusters. Despite having adjacencies to two clusters, the green node is not an overlap node.

To quantify the strength of a node's connection to multiple communities, different strategies could be applied. For example, in [46], overlap nodes are detected using a two-part criterion based not only on adjacencies, but also on a node's distance from the hubs of the corresponding communities. In [47], overlap is based on the "distance" from a node to a community, and it is determined with a heuristic that counts the number of triangles between a node and a community.

In this work, the Edge Dispersion Variation (EDV) measure is utilized to quantify how tightly a node u_x is connected to a given set Q of communities. The EDV for u_x, given a set of communities

$Q = \{C_1, C_2, ..., C_n\}$, is defined as a normalized standard deviation of the percentages of u_x's edges that link to Q's communities:

$$EDV(u_x, Q) = \sqrt{\sum_{i=1}^{|Q|} (a_{u_x,C_i} - \mu_Q)^2}, \tag{11}$$

where μ_Q is the mean of a_{u_x,C_i} computed over Q and a_{u_x,C_i} $(0 \le a_{u_x,C_i} \le 1)$ is the percentage of node u_x's edges adjacent to Q that are also adjacent to cluster i:

$$a_{u_x,C_i} = \frac{|\{(u,v) \in E | u = u_x, v \in C_i\}|}{|\{(u,v) \in E | u = u_x, v \in Q\}|}. \tag{12}$$

The measure of the strength of the connection correlates inversely with the EDV value. Nodes (like the green node in Figure 5) that are not overlaps will have a higher EDV value.

4.2. Computation of NBR Measures

The existing NP-hardness results for the resilience measures considered in this work include [48] for integrity (Ref. [49] for toughness, Ref. [50] for tenacity, Ref. [51] for scattering number and [52] for conductance). The approximation-hardness of unsmoothened vertex attack tolerance (UVAT) $\hat{\tau}(G) = \min_{S \subset V} \frac{|S|}{|V-S-C_{max}(V-S)|}$ under four separate, plausible computational complexity assumptions was established in [10,53]. VAT, integrity, toughness, tenacity and smoothed inverse scattering number are all minimization problems, whose objective functions involve similar calculations of component orders and numbers. Based on the approximation hardness of UVAT, the conjectured hardness of VAT and the previous NP-hardness results for the resilience measures, a greedy betweenness centrality (BC) based algorithm that is similar to the implementation of VAT-Clust presented in [6], was used to calculate the resilience measures. This greedy BC based heuristic algorithm is detailed in [10] and referred to as Greedy–BC.

The Greedy–BC algorithm takes graph $G = (V, E)$ and set-resilience measure function R as input. Let n denote the total number of nodes in G. The Greedy–BC algorithm is as follows:

```
1.  Initialize $G_0 = G$, $S_0 = \emptyset$, $r_j = \infty$, $r^{min} = \infty$, and $S^{min} = \emptyset$.
2.  For $j := 1$ to $n$ do:
    i.    $u_j = \max_{u \in V(G_{j-1})} BC(u)$,
    ii.   $S_j = S_{j-1} \cup \{u_j\}$, and $r_j = R(G, S_j)$,
    iii.  If $r_j < r^{min}$, then assign
              $r^{min} = r_j$, and $S^{min} = S_j$,
    iv.   $G_j = G_{j-1} - \{u_j\}$.
3.  Output $S^{min}$ as the critical attack set achieving minimum resilience, and $r^{min}$ as the minimum
    value achieved for the resilience measure $R$.
```

The overall time complexity of the Greedy–BC algorithm depends on the algorithm used to determine the node with the highest BC in each iteration. In general, if that algorithm takes time $O(B)$, then the overall time complexity of Greedy–BC may be expressed as $O(|V|(B + |E|))$. If restricted to exact, deterministic, sequential computational models, the fastest known algorithm to determine the node with the highest BC is Brandes's Fast Betweenness Centrality algorithm [54], which takes $O(|V||E|)$ time, yielding an overall Greedy–BC time complexity of $O(|V|^2|E|)$. As in [10], in the context of the work, we implemented Greedy–BC using Brandes's algorithm [54] and applied it to all graphs of size up to two thousand vertices. (for graphs with over two thousand vertices, we use a distributed algorithm [55]).

In [10], it is experimentally shown that Greedy–BC implemented using Brandes' betweenness centrality algorithm exhibits excellent empirical approximation bounds for VAT, integrity and tenacity on a representative sample of 24-node graphs, despite the general approximation hardness results for UVAT. In [6], hill climbing was applied to improve the approximation of the resilience measures; however, it is shown in [10] that the accuracies of Greedy–BC were not significantly improved by either 1D or 2D hill climbing, even for larger scale-free graphs up to thousands of nodes. The non-parametrized, practical NBR–Clust implementation employed in this work uses Greedy–BC without hill climbing and with unweighted betweenness centrality to *simultaneously* approximate VAT, integrity, tenacity, toughness, and scattering in $O(|V|^2|E|)$ time.

While Greedy–BC as implemented above yields high clustering accuracy for all graphs studied as well as an improvement in speed over hill-climbing based methods, it is still too slow for applications to very large graphs of tens or hundreds of thousands of nodes. Therefore, we have considered alternative approaches to tackle very large graphs, including parallelized approximations to betweenness centrality [55] and the use of adaptive approximations [56].

5. Experimental Results

In this section, we empirically evaluate NBR–Clust on varied datasets and networks to demonstrate the robustness of the algorithm. We quantify performance using the percentage accuracy metric, as well as whether, for a given dataset, the method accurately determines the optimal number of clusters (according to the ground truth). Percentage accuracy is defined as the percentage of nodes correctly clustered for a given dataset.

First, the efficacy of NBR clustering is established by clustering four real native network datasets [2,57–59] and comparing the accuracies obtained by NBR–Clust to two well-known algorithms: Girvan–Newman [2] and Louvain modularity [60]. The results of NBR–Clust are also demonstrated on multiple types of synthetically generated graphs with diverse properties including regular and irregular degree graphs such as LFR benchmark networks [61]. Larger networks of 10,000 and 100,000 nodes are also clustered.

We are very much interested in applications of NBR Clustering for datasets not natively in graph format. To examine that issue, we conduct a comparative analysis of NBR–Clust on sample datasets obtained from the UCI Machine Learning Repository [62] along with three other algorithms: k-means [20], Girvan–Newman [2] and spectral clustering [3]. We evaluate the performance of NBR–Clust on synthetic datasets generated as multi-dimensional Gaussian mixtures [63]. Finally, the robustness and effectiveness of NBR clustering methodology in identifying and dealing with noise and overlaps is evaluated and illustrated.

Although this paper's NBR–Clustering framework is implemented hierarchically, it is noted that in many cases high accuracy is achieved *in one shot* (without requiring multiple recursive iterations). This suggests there is potential for using some measures to determine a natural number of clusters in a graph when the number of clusters is not known a priori. When clustering has been completed in one step, the percentage accuracy is displayed in bold.

5.1. Real Data in Native Network Form

NBR–Clust using VAT, integrity, toughness, tenacity and scattering number was applied on four sets of data in native network form. The first network represents a set of books about US politics, compiled by Valdis Krebs [57]. The graph's nodes represent 105 books sold by Amazon (Seattle, WA 98121, USA), categorized by Mark Newman as liberal, neutral, or conservative. The edges represent frequent co-purchasing of books by the same buyers, as indicated by the *customers who bought this item also bought these other books* feature on Amazon. For comparison, Girvan–Newman results for this graph were obtained from [64], and Louvain modularity [60] was calculated using Gephi software (version 0.9.2, The Gephi Consortium, from gephi.org, accessed on 30 July 2018). As shown in

Table 1, VAT, integrity, toughness and tenacity all achieved accuracies of 80% or higher on this dataset. Both Girvan–Newman and Louvain modularity attained a clustering accuracy of 84%.

Table 1. Percent accuracy for real data in native graph form. Bold text indicates one-step clustering.

Dataset	VAT	Integrity	Toughness	Tenacity	Scattering
political books	**83**	**83**	**80**	**80**	**49**
karate club	**68**	**97**	**68**	**82**	**85**
college football	89	90	**53**	**53**	**61**
food web	**73**	**73**	**48**	**70**	**48**

The three additional datasets included Zachary's karate club [59], college football [2] and the Chesapeake Bay food web [58]. The accuracy results for these datasets are shown in Table 1. With the karate club, integrity achieves a high accuracy, while VAT and toughness are less successful. Like integrity, Girvan–Newman and Louvain both cluster the karate club at 97%. VAT and integrity achieve high accuracy with the football dataset, correctly classifying at least 89% of the teams according to conference. Girvan–Newman clusters this dataset at 90% accuracy, and Louvain clusters it at 91%. VAT and integrity are the most successful of all methods with the noisy and difficult food web dataset, where both correctly cluster 73% of the nodes. Girvan–Newman clusters the food web at 70%, and Louvain clusters it at 67% accuracy. Note also that, except in two cases with college football, all graphs are clustered by NBR–Clust in one iteration.

5.2. Synthetic Data in Native Network Form

5.2.1. Girvan–Newman Graphs

The Girvan–Newman graphs were generated as described in [2]. These graphs have 1024 nodes, with either 64, 32, 16 or 8 equal-sized clusters. The graphs have a regular degree structure, with each node's degree equal to half the cluster size. The strength of the community structure depends on the mixing coefficient, μ. Ten different graphs were randomly generated for different mixing coefficients from $\frac{1}{32}$ to $\frac{1}{8}$. However, for this type of graph, higher mixing coefficients are not effective as the resulting graphs are very well connected. Hence, many of the resilience measures do not return a useful attack set.

Nevertheless, these graphs are interesting as they have a very large number of clusters with a small set of members per cluster. For example, each cluster of the 64-cluster graphs contains only 16 members. The large number of clusters implies that either a large critical attack set or a large number of iterations is required. The results for the Girvan–Newman graphs are shown in Table 2, where # represents the number of clusters in the graph, and μ is the mixing factor. Each data point represents an average over 10 randomly generated graphs. As can be observed, NBR–Clust using any measure other than scattering number performs very well on these types of graphs. Even with the most difficult graphs, with the mixing coefficient $\mu = \frac{1}{8}$, the lowest performance (scattering) is 85% accuracy while all other NBR measures attain at least 92%. Note that tenacity was able to cluster every graph in only one iteration. This is due to the large size of the generated attack sets.

Table 2. Percent accuracy for NBR–Clust on Girvan–Newman graphs for varying mixing factors μ. Bold text indicates one-step clustering.

	VAT				**Integrity**				**Toughness**				**Tenacity**				**Scattering**			
μ / #	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{8}$
8	100	100	100	100	100	100	100	100	100	97	**97**	**99**	**100**	**97**	**97**	**99**	69	64	92	63
16	99	100	100	100	99	100	100	100	**100**	**98**	**94**	**96**	**100**	**98**	**94**	**96**	46	59	86	49
32	100	99	100	99	100	95	100	97	**98**	**97**	**96**	**95**	**98**	**97**	**96**	**95**	48	48	85	58
64	100	100	99	93	100	**100**	**96**	85	**100**	**96**	**94**	**92**	**100**	**96**	**94**	**92**	36	30	74	37

5.2.2. Scale-Free LFR Benchmark Networks of 1024 Nodes

An important generalized class of synthetic networks is the LFR networks [61]. Note that the networks that may be generated by the LFR model include regular degree Girvan–Newman networks as well as scale-free networks of various parametric settings. A series of LFR benchmark networks was generated as described in [61]. Each of these graphs has 1024 nodes. In contrast to the regular degree structure of the Girvan–Newman graphs, these are scale-free networks, generated using power law distributions for degree, where the degree sequence exponent = -2, and the community size distribution exponent = -1. These are similar to real complex networks, which "are known to be characterized by heterogeneous distributions of degree and community sizes" [65]. Fifty different networks were generated, each containing between 4 and 14 clusters, with the cluster size ranging from 50 to 200 nodes. The degree distributions were highly irregular, with a typical degree distribution ranging from 7 to 64. The average degree was 16. The mixing parameter varied from 0.05 to 0.25, where a higher mixing parameter implies less tightly bound communities, and therefore a more difficult graph to cluster.

The results for the scale-free LFR networks are shown in Table 3. The average percentage accuracy is presented over 10 separate randomly generated graphs. For graphs with the best-defined community structures (a mixing factor of $\mu = 5\%$), all measures except scattering number correctly classify more than 95% of the nodes. Even where $\mu = 20\%$, toughness and tenacity correctly classify over 85% of the nodes. For this class of graphs, all of the NBR measures, except for scattering, clustered the graphs in one step.

Table 3. Percent Accuracy for NBR–Clustering on LFR Networks of 1024 Nodes. Bold text indicates one-step clustering.

Measure	μ = 0.05	0.10	0.15	0.20	0.25
VAT	**96**	**82**	**71**	**67**	**48**
integrity	**97**	**92**	**84**	**74**	**61**
toughness	**97**	**95**	**90**	**85**	**55**
tenacity	**97**	**95**	**90**	**85**	**55**
scattering	44	19	18	17	17

5.2.3. Large LFR Networks

To demonstrate the feasibility of NBR–Clust on larger graphs, two series of LFR networks were generated (10,000 nodes and 100,000 nodes). These networks contain 30 to 40 clusters of varying cluster sizes, ranging from 100 to 500 nodes for the 10,000-node graphs, and 1000 to 5000 nodes for the 100,000-node graphs. The degree distributions are highly irregular, with a typical degree distribution ranging from 20 to 60. Different values of μ were used ranging from 0.01 to 0.10 for the 10,000-node graphs, and from 0.001 to 0.010 for the 100,000-node graphs. To determine if the number of edges affects the clustering accuracy, two variations of the 100,000-node graphs were tried. One set of graphs had approximately 275,000 edges, and one set had approximately 825,000 edges.

To cluster these larger graphs, Greedy–BC was executed in a distributed fashion using a graphics processing unit (GPU) betweenness centrality approximation algorithm called Hybrid–BC [55], as described in [66]. Results are shown in Table 4. In two cases, the approximation provided by the GPU algorithm was not sufficient to successfully complete clustering with VAT.

With LFR networks, higher mixing factors imply increased difficulty of clustering. Despite the fact that exact betweenness centrality values were not used, clustering accuracies are still quite high. As is shown in Table 4, 10,000-node graphs with μ less than 0.10 were clustered with at least 96% accuracy. To give some idea of execution time, graphs 1 to 5 were clustered in less than five minutes. The last and most difficult graph was clustered in approximately 35 minutes, with 92% accuracy.

Results for the 100,000-node graphs are also shown in Table 4. Clustering accuracy is high for both the high and low density graphs. Only two of the graphs were clustered with less than 97% accuracy. Clustering times varied a great deal, depending on μ and degree. Graphs 1–4 were clustered in less than 30 min, graph 5 took about 2.5 h, and graph 6 took approximately 9.5 h.

Table 4. Clustering Results for LFR Nets of 10,000 and 100,000 Nodes.

	10,000 Nodes				**100,000 Nodes**			
			Percent Accuracy				**Percent Accuracy**	
#	**Edges**	μ	**VAT**	**Integrity**	**Edges**	μ	**VAT**	**Integrity**
1	325,380	0.01	100	100	275,542	0.001	100	97
2	324,990	0.02	98	98	275,894	0.003	100	100
3	324,000	0.03	98	98	275,270	0.005	100	92
4	326,182	0.04	98	98	828,375	0.001	100	99
5	323,552	0.05	96	98	825,222	0.005	98	99
6	322,370	0.10	-	92	826,895	0.010	-	100

5.2.4. Attributed Networks with Communities

A model for generating attributed networks with communities is presented in [67]. This model is described as "similar to the BTER model" (described in Section 5.2.5), except that it uses the similarity of the vertex attributes to determine the inter-cluster edges, while BTER uses a scale-free distribution. The results in [67] suggest that the community structure of these graphs is most affected by the total number of edges in the graph. Fifty graphs were generated with 1000 nodes and 10 clusters (of varying sizes), varying the total number of edges to achieve a specified average degree. The high degree graphs have a strong community structure and should cluster with high accuracies. The low degree graphs should be more difficult to cluster.

Table 5 illustrates the results as an average over 10 multiple graphs generated with the same parameters and average degree. Integrity does especially well on this type of graph, averaging 86% accuracy on the most difficult graphs and 100% accuracy on the easiest. Tenacity clustered every graph in one step, while integrity clustered 48 out of 50 graphs in one step. Additionally, VAT clustered these graphs in an average of two steps or less.

Table 5. Percent accuracy for attributed networks with communities. Bold text indicates one-step clustering.

	Average Degree				
Measure	**40**	**30**	**20**	**15**	**10**
VAT	98	88	86	71	**73**
integrity	**100**	**99**	**97**	**95**	**86**
toughness	**98**	**92**	**75**	**84**	**68**
tenacity	**98**	**92**	**75**	**85**	**68**
scattering	67	62	46	30	23

5.2.5. BTER Benchmark

A BTER network [68] of 1899 nodes was generated using the FEASTPACK Matlab distribution [69] (version 1.2, Sandia National Laboratories, Albuquerque, NM, USA). In this network model, Erdős–Rényi [70] communities are created with sizes following a scale-free distribution and then connected with edges chosen according to the Chung and Lu model [71]. This graph is different from the other tested models because, in addition to having a scale-free degree distribution, the community sizes follow a scale-free distribution. It has 198 clusters. Of these, 47 clusters contain only three nodes, 31 clusters contain four nodes, and the largest cluster contains 101 nodes. The graph is not connected. A visualization of this network is shown in Figure 6. VAT, integrity and tenacity performed very well on this network with a minimum of 84% accuracy, and all but 5 out of 198 communities were successfully detected. VAT clustered this network at 86% accuracy, and all but 3 of 198 communities were successfully detected.

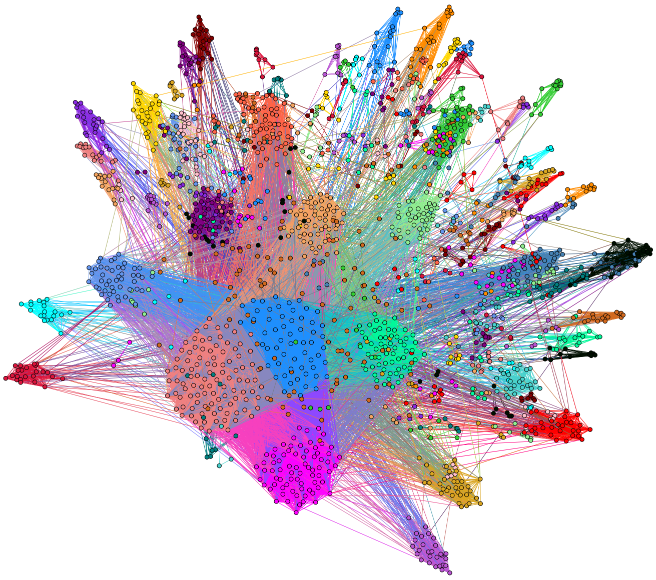


Figure 6. BTER Scale-Free network generated by FEASTPACK.

5.3. Real Data in Point Set Form

The results presented here were obtained by converting point set data into kNN graphs and clustering using NBR–Clust. Four real datasets from the UCI Machine Learning repository [62] are evaluated: iris, breast-WI, *E.Coli* and wine. Percent accuracy for these datasets is shown in Table 6. VAT, integrity and tenacity generally perform well on these datasets. The matching 90% accuracy of

VAT and integrity on the well-known iris dataset is particularly notable, as two of the three iris clusters are not linearly separable. For comparison to existing clustering methods (both point clustering and graph clustering), k-means clusters iris at 89%. The corresponding kNN graph is clustered by spectral clustering at 90% and by Girvan–Newman at 97%. Notably, integrity achieves a high accuracy on iris *in one shot*. Integrity and tenacity both clustered all four datasets in one shot.

k-means, spectral clustering, and Girvan–Newman clustered the breast-WI dataset at 85%, 82%, and 77% accuracy, respectively; while the NBR–Clust accuracies ranged from 81% to 85%. For *E. coli*, the highest accuracy is integrity at 60%, and most measures cluster closer to 55% accuracy. k-means clusters the *E. coli* dataset at 58% accuracy, spectral clustering at 50% and Girvan–Newman at 61% accuracy. On the wine dataset, NBR–Clust beat k-means and spectral, which clustered at 61% and 70% accuracy, and it tied with Girvan–Newman at 71% accuracy.

Table 6. Percent accuracy for NBR–Clustering on real point set data. Bold text indicates one-step clustering.

Dataset	VAT	Integrity	Toughness	Tenacity	Scattering
iris	90	**90**	51	**88**	51
breast-WI	**81**	**85**	**85**	**85**	**81**
E. coli	55	**60**	**53**	**53**	58
wine	71	**71**	71	**70**	71

5.4. Synthetic Data in Point Set Form

In this section, the performance of NBR–Clust on a set of synthetic datasets generated by Arbelaitz et al. [63] is evaluated. These datasets were generated as multi-dimensional Gaussian mixtures with either 10% uniformly random noise or no noise. The dimension D (the number of features of the point set) and the number of clusters K both vary as 2, 4, or 8. There are nine combinations of D and K, shown as one combination per row in Table 7. Arbelaitz et al. use the term density to describe whether or not the clusters of a dataset are equal in size. In the equal density datasets, each cluster contains 100 members. In the unequal density datasets, one cluster has 400 members, and the remaining clusters have 100 members. For example, a D4K8 equal density dataset has 800 members with no noise, and the corresponding dataset with 10% noise has 880 members. The unequal density D4K8 dataset will have 1100 members with no noise and 1210 members with noise. The Arbelaitz data contains 10 different generated sets of all 9 combinations of D and K, equal and unequal density, noisy and without noise, for a total of 360 datasets.

Table 7. Percent accuracy for four variations of synthetic datasets. Bold text indicates one-step clustering.

		VAT				**Integrity**				**Tenacity**			
		no Noise		**with Noise**		**no Noise**		**with Noise**		**no Noise**		**with Noise**	
D	**K**	**eq dens**	**uneq dens**	**eq dens**	**uneq dens**	**eq dens**	**uneq dens**	**eq dens**	**uneq dens**	**eq dens**	**uneq dens**	**eq dens**	**uneq dens**
2	2	**100**	**100**	**100**	**96**	**100**	**100**	**49**	**80**	**100**	**100**	**52**	**81**
2	4	99	100	100	100	**99**	**99**	**36**	**69**	**99**	**99**	**28**	**57**
2	8	100	100	98	97	**100**	**100**	**76**	**78**	**100**	**100**	**25**	**36**
4	2	**100**	**100**	**100**	**100**	**100**	**100**	**51**	**80**	**100**	**100**	**52**	**80**
4	4	100	100	100	99	**100**	100	**43**	**79**	**100**	**99**	**26**	**57**
4	8	100	100	99	97	**100**	100	**97**	**96**	**100**	**99**	**14**	**36**
8	2	**100**	**100**	**100**	**100**	**100**	**100**	**50**	**80**	**100**	**100**	**51**	**80**
8	4	100	100	95	100	**100**	100	**71**	**89**	**100**	**100**	**34**	**57**
8	8	99	100	98	97	**99**	100	**100**	**95**	**99**	**100**	**14**	**36**

5.4.1. Synthetic Point Set NBR–Clust Results

The results for clustering the synthetic point set datasets are shown in Table 7 as an average percent accuracy over the 10 instances per dataset. Toughness and scattering generally performed poorly, hence only the results for VAT, integrity, and tenacity are presented.

VAT attained a high accuracy on all four types of synthetic point set graphs. VAT clustered almost all of the noiseless graphs perfectly, achieving a 99% or 100% average accuracy on each series of 10 graphs. VAT did only slightly worse on the noisy graphs, clustering both equal and unequal density graphs at 95% or above. In general, VAT did not cluster in one step. Clustering took between 1 and 7 steps, and the number of steps did not vary greatly between the four data set variations.

Like VAT, integrity clustered the noiseless datasets almost perfectly, achieving an accuracy of 99% or higher. On the noisy unequal density datasets, integrity did relatively well, clustering between 69% and 95% accuracy. Integrity was less successful with the noisy datasets, although it exhibited a relatively consistent pattern of increasing accuracy as the number of clusters increased, independent of the number of dimensions. Integrity clustered every noisy dataset, and most of the noiseless datasets, in only one iteration.

Tenacity also clustered the non-noisy datasets with high accuracy. Tenacity did not perform as well on the noisy datasets. In contrast to integrity, tenacity performed better on datasets with a smaller number of clusters. Tenacity clustered every dataset in only one step, without requiring further recursive iterations.

5.5. Noise Removal

This section specifically evaluates the performance of NBR clustering in removing noise from noisy synthetic datasets, in contrast to the previous section, which presented percentage accuracy results on NBR–Clust's effectiveness in clustering noisy datasets. The Arbelaitz point set data is utilized, which contains 10% noise that is randomly generated following a "uniform distribution over the sampling window" [63]. A visualization of an unequal density dataset with noise is illustrated in Figure 7. The cases of primary interest are those where NBR–Clust successfully removes noise without compromising clustering accuracy, and it is preferred if this is accomplished in one iteration.

The measures precision and recall are used to quantify NBR–Clust performance when identifying noise nodes. Let N_D denote the number of noise nodes in the dataset; $|S|$, the number of nodes in the critical attack set; and N_S, the number of noise nodes contained in that attack set. Precision is defined as P = $\frac{N_S}{|S|}$, and recall is defined as RC = $\frac{N_S}{N_D}$. Precision indicates how successful the algorithm was in identifying the noise nodes as the attack set nodes, while recall is the percentage of the overall noise nodes that were contained in the attack set.

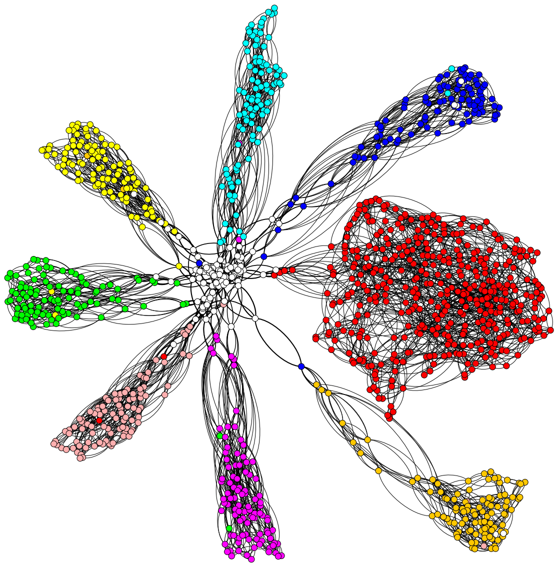


Figure 7. Visualization of a low degree graph created from a D8K8 dataset. Different colors represent different ground-truth clusters. White nodes are noise (note the large number of noise nodes at the center of the graph).

Table 8 shows the precision and recall of NBR–Clust using VAT, integrity and tenacity when removing noise from noisy datasets. Integrity gives the most balanced mix of precision and recall. VAT generally has a high precision—the nodes it finds are quite likely to be noise nodes, but due to the smaller attack set sizes, it does not find a large percentage of noise nodes. Tenacity has the opposite problem, often finding over 90% of noise nodes, but also with the high probability that many of them are not noise, implying low precision but high recall.

Table 8. Precision and Recall (in Percentage) of Hierarchical NBR Clustering Noise Removal on Noisy Synthetic Datasets.

		VAT				**Integrity**				**Tenacity**			
		eq dens		**uneq dens**		**eq dens**		**uneq dens**		**eq dens**		**uneq dens**	
D	**K**	**Pre**	**Rec**	**Pre**	**Rec**	**Pre**	**Rec**	**Pre**	**Rec**	**Pre**	**Rec**	**Pre**	**Rec**
2	2	63	6	65	3	25	40	40	57	17	93	19	96
2	4	58	7	68	5	22	24	36	46	16	91	15	91
2	8	74	14	74	10	53	28	34	21	15	81	15	84
4	2	80	8	89	8	26	68	30	77	17	94	17	98
4	4	83	13	93	11	30	57	17	96	37	72	17	96
4	8	82	16	85	13	96	49	47	51	15	95	16	95
8	2	80	10	75	5	26	75	27	75	18	94	19	95
8	4	85	18	90	16	38	45	32	69	17	88	17	95
8	8	92	22	94	23	96	51	44	63	16	93	17	93

6. Clustering with Overlap

Evaluating the quality of the clustering configuration obtained in the presence of overlaps is difficult. It requires a metric that accounts for the number of nodes correctly assigned to clusters, as well as for false and missing overlap assignments. Only a few metrics are available for evaluating overlapping communities [33]. The commonly used normalized mutual information (NMI) was enhanced specifically for overlapping communities in [72] and further enhanced with more intuitive normalization in [73]. To calculate NMI, the intuitive algorithm described by McDaid et al. in [73] and the software implementation provided on their website were used.

For detecting overlap nodes using EDV (previously described with the NBR clustering framework), the value of 0.3 was used as the threshold standard deviation. This was derived from the optimal value obtained from preliminary experiments on varying the threshold value and observing the number of critical set nodes detected while clustering complex datasets with high accuracy.

6.1. Results with Overlap Detection

6.1.1. Girvan–Newman Graphs with Overlap

This paper's overlapping community detection algorithm was tested with NBR–Clust using VAT and integrity on a series of Girvan–Newman style graphs. These graphs were generated in the same fashion as described in Section 5.2.1. Each graph has 256 nodes, with 32 overlap nodes, meaning that a high percentage of the nodes (one-eighth) belong to more than one cluster. Each overlap node is a member of two, four or six communities. The number of communities that a node overlaps is consistent within a graph. For each combination of two, four, and six overlapped communities, mixing factor was varied from $\frac{1}{32}$, which should be easiest to cluster, to $\frac{1}{8}$, which should be the most difficult. The results for these experiments are shown in Table 9, where μ denotes the mixing factor, # the number of memberships of an overlap node, and NMI values represent the average over 10 randomly generated graphs.

Table 9. Normalized mutual information for Girvan–Newman Graphs with overlap. Bold text indicates one-step clustering.

	VAT				Integrity			
μ / #	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{3}{32}$	$\frac{1}{8}$
2	1.00	0.92	0.99	0.84	**1.00**	0.81	0.94	0.86
4	0.98	0.94	0.94	0.82	**0.99**	0.88	0.92	0.81
6	0.94	0.87	0.87	0.83	**0.95**	0.85	0.86	0.81

Both integrity and VAT were able to cluster the two-community overlap low-mixing factor graphs perfectly, finding every overlap node and assigning each to the correct clusters. Clustering where nodes overlap six communities is a much more difficult problem, and VAT still managed to achieve an NMI of 0.94 on the low-mixing factor graphs and 0.83 on the graphs with a higher mixing factor. Integrity did about the same, scoring 0.95 on the low-mixing factor graphs and 0.81 on the high-mixing factor graphs. Since misidentified nodes are assigned to multiple incorrect communities, these NMI numbers imply that less than 2% of the nodes were incorrectly identified, even in the worst cases.

Most clustering took more than one iteration. Only the $\mu = \frac{1}{32}$ graphs were clustered by integrity in one step. VAT generally took four to seven steps to cluster, with a maximum of 10 steps. Integrity took between one and six steps. The number of steps increased with the number of communities per overlap node, regardless of the mixing factor.

6.1.2. LFR Benchmark Networks with Overlap

LFR benchmark networks were generated with overlap as described in [61]. All graphs have 1024 nodes, and are scale-free with a degree sequence exponent of -2 and a community size distribution exponent of -1. The mixing factor is 0.1, and the cluster size is not fixed, but ranges from 50 to 200 nodes. Each graph has a fixed number of overlap nodes and a constant number of memberships for each overlap node. The number of overlap nodes were tested at 25, 50 and 100, and the number of memberships for each overlap node were tested at 2, 4 and 6.

The results of the clustering and reassigning are shown in Table 10. On the easiest graphs, VAT scored an NMI of 0.98, and integrity scored 0.93. Even with the very most difficult clustering problem, 100 nodes overlapping six different clusters, VAT and integrity got NMI scores of 0.88 and 0.84, respectively. These scores imply that less than 2% of the nodes were incorrectly identified. Additionally, integrity was able to cluster each type of graph in a median of one step. VAT clustered in a median of three steps.

Table 10. Normalized mutual information for scale free LFR benchmark networks with overlap. Bold text indicates one-step clustering.

	VAT			**Integrity**		
	# Overlap Nodes			**# Overlap Nodes**		
Memberships	**25**	**50**	**100**	**25**	**50**	**100**
2	0.98	0.97	0.98	**0.93**	**0.92**	**0.96**
4	0.97	0.96	0.95	**0.97**	**0.97**	**0.95**
6	0.96	0.93	0.88	**0.96**	**0.93**	**0.84**

7. Discussion

The theoretical motivation for a generalized NBR clustering framework is based on the relationship between resilience and clustering. This relationship is explicit in the sparsest cuts clustering framework, and has been extended in a novel manner to NBR measures, which presents unique challenges as well as advantages for noise removal and overlapping community detection. This paper's theoretical motivation is strengthened by defining an infinite family of graphs, as illustrated in Figure 2, such that using any node-based resilience measure for clustering immediately (in one iteration) yields the correct partitions. Furthermore, a generalized node-based resilience clustering framework (NBR–Clust) is presented that is naturally applicable for variations of the traditional clustering problem in which a complete partitioning is not appropriate, such as clustering in the presence of noise as well as in the presence of overlapping clusters.

The primary goal of this work has been to demonstrate the effectiveness of NBR–Clustering methodology via extensive experimental results across variations of the clustering problem and a diverse set of datasets, including real and synthetic, as well as native graph data and point set data. There are computational hardness results pertaining to all resilience measures considered in this research's NBR–Clustering framework. The heuristic, $O(|V|^2|E|)$ time Greedy–BC method is utilized to *simultaneously* compute all relevant measures given the acceptable empirical approximation factors attained for that approach in [10] as well as the super-linear improvement in the time complexity of Greedy–BC over Greedy–BC with hill-climbing. Additionally, use of a parallelized algorithm as described in [55] allowed the clustering of larger graphs of up to 100,000 nodes in reasonable times. In the context of this work, the main concern is the quality of the clusterings obtained via NBR–Clustering as well as examination of NBR measures and data scenarios for which NBR–Clustering performs (i) accurate one-iteration clustering without the number of clusters known a priori; (ii) noise removal with high precision and recall and (iii) high-quality detection of overlapping communities.

NBR–Clust exhibited competitive results compared to the popular Girvan–Newman and Louvain community detection methods on a series of well-known real networks, outperforming those methods on the food web dataset and tying them on the karate club network. Regarding performance on synthetic graph data, this paper has demonstrated that NBR–Clust gives robust results for both regular and scale-free degree distributions, for varying cluster size distributions, across a reasonable range of mixing factors, and even on complex networks with attributes. NBR–Clustering results on real and synthetic datasets generated from point-set data transformed into kNN graphs have been similarly promising, both in noiseless and noisy (10% uniformly random noise) settings. On a series of kNN graphs generated from Gaussian-mixture point-set data, VAT, integrity and tenacity clustered noiseless datasets almost perfectly, with VAT also clustering all noisy point-set data almost perfectly. VAT also resulted in accurate clustering for the BTER complex network dataset. Although the BTER network's scale-free distribution of both cluster size and degree is known to cause difficulty for many clustering algorithms, NBR–Clustering, using VAT, was able to detect all but 3 of 198 clusters.

While VAT has been particularly notable in accurate clustering across various datasets, integrity and tenacity have shown promise in *one step* clustering across a variety of network and point set data when the number of clusters is unknown a priori. Both integrity and tenacity clustered all attributed network datasets in a single iteration, with integrity clustering at an average of 86% accuracy. For the LFR benchmarks, attributed networks with communities and synthetic point set graphs, integrity clustered 42 out of 46 sets (of 10 graphs) in only one step. Tenacity clustered every regular Girvan–Newman graph in one step with an average accuracy greater than 92%. Tenacity also clustered every kNN graph generated from noiseless and noisy synthetic point set data in one step, and integrity clustered the majority of such data sets in one step with relatively higher accuracies. Upon closer examination of the poor performance of tenacity on noisy synthetic datasets, it was observed that many more nodes were removed by tenacity (including many non-noise nodes), resulting in a higher number of clusters than necessary. Thus, even in cases where tenacity does not achieve an accurate clustering in one step, the number of clusters formed by tenacity serves as an upper bound on the correct number of clusters desired. Finally, in the context of clustering in a single iteration, this research shows that, for *all measures* except scattering, NBR–Clust clustered almost every scale-free LFR benchmark network in a single iteration.

Another important contribution of this work is the application to noise removal. The efficacy of NBR–Clust has been examined with respect to noise removal which occurs as an automatic by-product of node-based resilience measure computation, particularly computation of the critical attack set, which is taken as a candidate set of noise nodes. As stated previously, the noisy datasets of this work are minimum connectivity kNN graphs with Gaussian mixture ground truth clusters and an additional 10% uniformly random noise. VAT showed high precision in detecting noise nodes, with the percentage of noise nodes in the critical attack sets ranging from 63% to 94%. VAT's small attack set size implied that the total number of noise nodes found was small. In contrast, tenacity generally detected over 90% of the noise nodes, but with low precision. Integrity had the best combination of precision and recall with noise detection, although results varied. Noise removal with one-shot integrity-based partial clustering was particularly notable on two classes of noisy point sets: equal density D4K8 and equal density D8K8. For the D4K8 datasets, integrity identified approximately 50% of the noise nodes, with a precision of 96%, while still retaining one-shot clustering accuracies of 97% and 100%. It is interesting to consider the graph corresponding to the noisy, D8K8 unequal density dataset shown in Figure 7. The central location of the noise nodes in this scenario yields a natural stochastic generalization of this paper's illustrative example motivating node-based resilience measures for partial clustering in Figure 2.

It is not necessary for the noise to form a single centrally located group in order for NBR clustering to cluster effectively. For example, consider the noisy, equal density D2K8 dataset shown in Figure 3.

Attack-set noise nodes are shown as triangles, and the corresponding min-conn kNN graph is shown in Figure 4. All NBR measures, except scattering, clustered the graph perfectly. The noise removal on that dataset was also at least 49% for all resilience measures considered. It can be observed that the uniformly at random distributed noise still tends to form high-betweenness nodes "between" natural clusters.

Finally, this paper discusses the effectiveness of NBR–Clust with respect to overlapping communities. There are relatively few algorithms that can account for overlap. The hypothesis motivating the application of node-based resilience for this setting was that overlap nodes would be a subset of attack set nodes because they both lie on inter-cluster boundaries. This hypothesis has been strongly supported by this paper's experiments as the attack sets contained, on average, at least 99% of the overlap nodes across all datasets with overlapping communities.

On regular Girvan–Newman graphs with overlapping communities, NBR–Clust results with VAT and integrity have been promising for lower mixing factors, as both measures have clustered such graphs with at least 0.94 NMI. Amongst the Girvan–Newman graphs with overlap, the most difficult clustering situations had 32 of 256 nodes overlapping 6 out of 13 clusters, and VAT still achieved an NMI of 0.83, while integrity scored 0.81. The results are similarly notable for integrity on scale-free benchmark graphs with overlap. Integrity achieved high NMI scores while clustering every graph in one step. In the case where 100 nodes overlapped six clusters each, integrity scored an NMI of 0.84. In this case, an average of less than 2% of the nodes were misidentified (and incorrectly assigned to clusters).

As far as assignment of overlap nodes is concerned, one of the problems of overlap clustering algorithms in general is that the overlaps make distinct clusters difficult to determine. Approaches such as maximizing modularity must consider many different options and do not have deterministic outcomes. One of the benefits of NBR–Clust is that the initial clusters are both disjoint (because overlaps have been removed as part of the attack set) and reflective of the natural number of clusters of the graph, so that critical node assignment can be a low time complexity, deterministic process. Due to such inherent properties of critical attack sets for node-based resilience measures, this research demonstrates that a straightforward adjacency-based approach yields a successful strategy to subsequently assign the overlapping nodes to multiple clusters after the critical attack set computation.

8. Conclusions

A generalized NBR clustering framework has been presented and several important facets of this novel algorithm have been demonstrated. First, with appropriate heuristics, it can be accomplished practically and without sacrificing accuracy despite the inherent hardness of the resilience measures. Second, NBR clustering in general, and integrity and tenacity in particular, demonstrate an ability to correctly determine the number of clusters present in datasets where that number is not given a priori. Even when tenacity does not produce an accurate clustering, the number of resulting components it produces is a useful upper bound on the desired number of clusters. Third, when node-based resilience clustering methods are applied to noisy datasets, the removal of critical attack set nodes also corresponds to the removal of noise. Lastly, overlap nodes are almost always part of the attack set selected by each resilience measure, and the subsequent assignment of clusters to the overlapping nodes can be accomplished successfully by a straightforward adjacency-based approach due to the high quality of the initial overlap detection. Overall, this work demonstrates the robustness of this technique across a variety of clustering situations.

Supplementary Materials: The following are available online at http://www.cs.siue.edu/~gercal/clustering/.

Author Contributions: J.M. and G.E. developed the algorithmic framework and ran the experiments. J.B. wrote the original VAT-based clustering software which J.M. expanded significantly into the generalized NBR–Clust with applications to noisy and overlapping datasets. K.S., D.W. and T.O.-A. contributed to discussions and ideas. J.M., G.E. and T.O.-A. shared the writing. All authors read and approved the final manuscript.

Funding: This research received no external funding.

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

Abbreviations

The following abbreviations are used in this manuscript:

BC	Betweenness Centrality
BTER	Block Two-Level Erdös–Rényi
EDV	Edge Dispersion Variation
kNN	k-Nearest Neighbors
LFR	Lancichinetti–Fortunato–Radicchi
NBR–Clust	Node Based Resilience Clustering
NMI	Normalized Mutual Information
UVAT	Unsmoothened Vertex Attack Tolerance
VAT	Vertex Attack Tolerance

References

1. Fortunato, S. Community detection in graphs. *Phys. Rep.* **2010**, *486*, 75–174. [CrossRef]
2. Girvan, M.; Newman, M.E. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [CrossRef] [PubMed]
3. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.
4. Alpert, C.J.; Kahng, A.B.; Yao, S.Z. Spectral partitioning with multiple eigenvectors. *Discret. Appl. Math.* **1999**, *90*, 3–26. [CrossRef]
5. Chung, F. *Spectral Graph Theory*; American Mathematical Society: Providence, RI, USA, 1997.
6. Borwey, J.; Ahlert, D.; Obafemi-Ajayi, T.; Ercal, G. A Graph-Theoretic Clustering Methodology Based on Vertex-Attack Tolerance. In Proceedings of the The Twenty-Eighth International Flairs Conference, Hollywood, FL, USA, 18–20 May 2015.
7. Ercal, G.; Matta, J. Resilience Notions for Scale-free Networks. *Procedia Comput. Sci.* **2013**, *20*, 510–515. [CrossRef]
8. Matta, J.; Borwey, J.; Ercal, G. Comparative Resilience Notions and Vertex Attack Tolerance of Scale-Free Networks. *arXiv* **2014**, arXiv:1404.0103.
9. Ercal, G. On Vertex Attack Tolerance of Regular Graphs. *arXiv* **2014**, arXiv:1409.2172.
10. Matta, J.; Ercal, G.; Borwey, J. The vertex attack tolerance of complex networks. *RAIRO-Oper. Res.* **2017**, *51*, 1055–1076. [CrossRef]
11. Barefoot, C.; Entringer, R.; Swart, H. Vulnerability in graphs-a comparative survey. *J. Comb. Math. Comb. Comput.* **1987**, *1*, 12–22.
12. Chvatal, V. Tough graphs and hamiltonian circuits. *Discret. Math.* **2006**, *306*, 910–917. [CrossRef]
13. Cozzens, M.; Moazzami, D.; Stueckle, S. The tenacity of a graph. In Proceedings of the Seventh International Conference on the Theory and Applications of Graphs, Kalamazoo, MI, USA, 1–5 June 1992; Wiley: New York, NY, USA, 1995; pp. 1111–1122.
14. Jung, H. On maximal circuits in finite graphs. *Ann. Discrete Math.* **1978**, *3*, 129–144.
15. Matta, J.; Obafemi-Ajayi, T.; Borwey, J.; Wunsch, D.; Ercal, G. Robust Graph-Theoretic Clustering Approaches Using Node-Based Resilience Measures. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, 12–15 December 2016; pp. 320–329. [CrossRef]
16. Matta, J.; Nguyen, T.; Ercal, G.; Obafemi-Ajayi, T. Applications of Novel Graph Theoretic Methods to Clustering Autism Spectrum Disorders Phenotypes. In Proceedings of the International Conference on Bioinformatics and Computational Biology (BICOB), Honolulu, HI, USA, 20–22 March 2017.
17. Elhaik, E.; Yusuf, L.; Anderson, A.I.; Pirooznia, M.; Arnellos, D.; Vilshansky, G.; Ercal, G.; Lu, Y.; Webster, T.; Baird, M.L.; et al. The Diversity of REcent and Ancient huMan (DREAM): A new microarray for genetic anthropology and genealogy, forensics, and personalized medicine. *Genome Biol. Evol.* **2017**, *9*, 3225–3237. [CrossRef] [PubMed]

18. Dale, J.; Matta, J.; Howard, S.; Ercal, G.; Qiu, W.; Obafemi-Ajayi, T. Analysis of Grapevine Gene Expression Data using Node-Based Resilience Clustering. In Proceedings of the 2018 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, St. Louis, MO, USA, 30 May–2 June 2018.
19. Cukierski, W.J.; Foran, D.J. Using betweenness centrality to identify manifold shortcuts. In Proceedings of the 2008 IEEE International Conference on Data Mining Workshops (ICDMW'08), Pisa, Italy, 15–19 December 2008; pp. 949–958.
20. Xu, R.; Wunsch, D. *Clustering*; Wiley-IEEE Press: Hoboken, NJ, USA, 2009.
21. Arora, S.; Rao, S.; Vazirani, U.V. Expander flows, geometric embeddings and graph partitioning. *J. ACM* **2009**, *56*, 5. [CrossRef]
22. Chawla, S.; Krauthgamer, R.; Kumar, R.; Rabani, Y.; Sivakumar, D. On the Hardness of Approximating Multicut and Sparsest-Cut. *Comput. Complex.* **2006**, *15*, 94–114. [CrossRef]
23. Newman, M.E. Detecting community structure in networks. *Eur. Phys. J. B* **2004**, *38*, 321–330. [CrossRef]
24. Bouhali, S.; Ellouze, M. Community detection in social network: Literature review and research perspectives. In Proceedings of the 2015 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), Hammamet, Tunisia, 15–17 November 2015; pp. 139–144.
25. Newman, M.E. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **2004**, *69*, 066133. [CrossRef] [PubMed]
26. Hawkins, D.M. *Identification of Outliers*; Springer: Berlin, Germany, 1980; Volume 11.
27. Frénay, B.; Verleysen, M. Classification in the presence of label noise: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 845–869. [CrossRef] [PubMed]
28. Ott, L.; Pang, L.; Ramos, F.T.; Chawla, S. On integrated clustering and outlier detection. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Dutchess County, NY, USA, 2014; pp. 1359–1367.
29. Chawla, S.; Gionis, A. k-Means—A unified approach to clustering and outlier detection. In *Proceedings of the 2013 SIAM International Conference on Data Mining*; Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, USA, 2013; pp. 189–197.
30. Obafemi-Ajayi, T.; Lam, D.; Takahashi, T.N.; Kanne, S.; Wunsch, D. Sorting the phenotypic heterogeneity of autism spectrum disorders: A hierarchical clustering model. In Proceedings of the 2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Niagara Falls, ON, Canada, 12–15 August 2015; pp. 1–7.
31. Charikar, M.; Khuller, S.; Mount, D.M.; Narasimhan, G. Algorithms for facility location problems with outliers. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2001; pp. 642–651.
32. McCutchen, R.M.; Khuller, S. Streaming algorithms for k-center clustering with outliers and with anonymity. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*; Springer: Berlin, Germany, 2008; pp. 165–178.
33. Xie, J.; Kelley, S.; Szymanski, B.K. Overlapping Community Detection in Networks: The State-of-the-art and Comparative Study. *ACM Comput. Surv.* **2013**, *45*, 43. [CrossRef]
34. Arora, S.; Ge, R.; Sachdeva, S.; Schoenebeck, G. Finding Overlapping Communities in Social Networks: Toward a Rigorous Approach. In Proceedings of the 13th ACM Conference on Electronic Commerce (EC'12), Valencia, Spain, 4–8 June 2012; ACM: New York, NY, USA, 2012; pp. 37–54. [CrossRef]
35. Derényi, I.; Palla, G.; Vicsek, T. Clique percolation in random networks. *Phys. Rev. Lett.* **2005**, *94*, 160202. [CrossRef] [PubMed]
36. Ahn, Y.Y.; Bagrow, J.P.; Lehmann, S. Link communities reveal multiscale complexity in networks. *Nature* **2010**, *466*, 761–764. [CrossRef] [PubMed]
37. Baumes, J.; Goldberg, M.K.; Krishnamoorthy, M.S.; Magdon-Ismail, M.; Preston, N. Finding communities by clustering a graph into overlapping subgraphs. *IADIS AC* **2005**, *5*, 97–104.
38. Sinclair, A.; Jerrum, M. Approximate Counting, Uniform Generation and Rapidly Mixing Markov Chains. *Inf. Comput.* **1989**, *82*, 93–133. [CrossRef]
39. Kajdanowicz, T.; Morzy, M. Using Graph and Vertex Entropy to Compare Empirical Graphs with Theoretical Graph Models. *Entropy* **2016**, *18*, 320. [CrossRef]
40. Ai, X. Node Importance Ranking of Complex Networks with Entropy Variation. *Entropy* **2017**, *19*, 303. [CrossRef]
41. Berge, C. *Hypergraphs: Combinatorics of Finite Sets*; Elsevier: New York, NY, USA, 1984; Volume 45.

42. Roy, S.; Ravindran, B. Measuring network centrality using hypergraphs. In Proceedings of the Second ACM IKDD Conference on Data Sciences, Bangalore, India, 18–21 March 2015; ACM: New York, NY, USA, 2015; pp. 59–68.
43. Node-Based Resilience Measure Clustering Project Website. Available online: http://www.cs.siue.edu/~gercal/clustering/ (accessed on 30 July 2018).
44. Maier, M.; Luxburg, U.V.; Hein, M. Influence of graph construction on graph-based clustering measures. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Dutchess County, NY, USA, 2008; pp. 1025–1032.
45. Yang, J.; Leskovec, J. Defining and evaluating network communities based on ground-truth. *Knowl. Inf. Syst.* **2015**, *42*, 181–213. [CrossRef]
46. Liu, W.; Pellegrini, M.; Wang, X. Detecting communities based on network topology. *Sci. Rep.* **2014**, *4*, 5739. [CrossRef] [PubMed]
47. Lyu, T.; Bing, L.; Zhang, Z.; Zhang, Y. Efficient and Scalable Detection of Overlapping Communities in Big Networks. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, 12–15 December 2016; pp. 1071–1076.
48. Drange, P.G.; Dregi, M.S.; Van't Hof, P. On the computational complexity of vertex integrity and component order connectivity. In *Algorithms and Computation*; Springer International Publishing: Basel, Switzerland, 2014; pp. 285–297.
49. Bauer, D.; Hakimi, S.L.; Schmeichel, E. Recognizing tough graphs is NP-hard. *Discret. Appl. Math.* **1990**, *28*, 191–195. [CrossRef]
50. Mann, D.E. The Tenacity of Trees. Ph.D. Thesis, Northeastern University, Boston, MA, USA, 1993.
51. Broersma, H.; Fiala, J.; Golovach, P.A.; Kaiser, T.; Paulusma, D.; Proskurowski, A. Linear-Time Algorithms for Scattering Number and Hamilton-Connectivity of Interval Graphs. *J. Graph Theory* **2015**, *79*, 282–299. [CrossRef]
52. Šíma, J.; Schaeffer, S.E. On the NP-completeness of some graph cluster measures. In *SOFSEM 2006: Theory and Practice of Computer Science*; Springer: Berlin, Germany, 2006; pp. 530–537.
53. Ercal, G. A Note on the Computational Complexity of Unsmoothened Vertex Attack Tolerance. *arXiv* **2016**, arXiv:1603.08430.
54. Brandes, U. A Faster Algorithm for Betweenness Centrality. *J. Math. Sociol.* **2001**, *25*, 163–177. [CrossRef]
55. McLaughlin, A.; Bader, D.A. Scalable and high performance betweenness centrality on the GPU. In Proceedings of the International Conference for High performance Computing, Networking, Storage and Analysis, New Orleans, LA, USA, 16–21 November 2014; IEEE Press: Piscataway, NJ, USA, 2014; pp. 572–583.
56. Yoshida, Y. Almost linear-time algorithms for adaptive betweenness centrality using hypergraph sketches. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 1416–1425.
57. Krebs, V. Books about US Politics. Available online: http://www.orgnet.com (accessed on 30 July 2018).
58. Baird, D.; Ulanowicz, R.E. The seasonal dynamics of the Chesapeake Bay ecosystem. *Ecol. Monogr.* **1989**, *59*, 329–364. [CrossRef]
59. Zachary, W.W. An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **1977**, *33*, 452–473. [CrossRef]
60. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
61. Lancichinetti, A.; Fortunato, S.; Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **2008**, *78*, 046110. [CrossRef] [PubMed]
62. Frank, A.; Asuncion, A. UCI Machine Learning Repository. 2010. Available online: http://archive.ics.uci.edu/ml (accessed on 30 July 2018).
63. Arbelaitz, O.; Gurrutxaga, I.; Muguerza, J.; Pérez, J.M.; Perona, I.N. An Extensive Comparative Study of Cluster Validity Indices. *Pattern Recognit.* **2013**, *46*, 243–256. [CrossRef]
64. Zhao, P.; Zhang, C.Q. A new clustering method and its application in social networks. *Pattern Recognit. Lett.* **2011**, *32*, 2109–2118. [CrossRef]
65. Lancichinetti, A.; Fortunato, S. Community detection algorithms: A comparative analysis. *Phys. Rev. E* **2009**, *80*, 056117. [CrossRef] [PubMed]

66. Matta, J. A Comparison of Approaches to Computing Betweenness Centrality for Large Graphs. In Proceedings of the International Workshop on Complex Networks and Their Applications, Lyon, France, 29 November–1 December; Springer: Cham, Switzerland, 2017; pp. 3–13.
67. Largeron, C.; Mougel, P.N.; Rabbany, R.; Zaïane, O.R. Generating attributed networks with communities. *PLoS ONE* **2015**, *10*, e0122777. [CrossRef] [PubMed]
68. Kolda, T.G.; Pinar, A.; Plantenga, T.; Seshadhri, C. A scalable generative graph model with community structure. *SIAM J. Sci. Comput.* **2014**, *36*, C424–C452. [CrossRef]
69. Kolda, T.G.; Pinar, A. et al. *FEASTPACK v1.2*; Sandia National Laboratories, Albuquerque, NM, USA 2014.
70. Erdös, P.; Rényi, A. On random graphs, I. *Publ. Math.* **1959**, *6*, 290–297.
71. Chung, F.; Lu, L. The average distances in random graphs with given expected degrees. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 15879–15882. [CrossRef] [PubMed]
72. Lancichinetti, A.; Fortunato, S.; Kertész, J. Detecting the overlapping and hierarchical community structure in complex networks. *New J. Phys.* **2009**, *11*, 033015. [CrossRef]
73. McDaid, A.F.; Greene, D.; Hurley, N. Normalized Mutual Information to evaluate overlapping community finding algorithms. *arXiv* **2011**, arXiv:1110.2515.