The Journal of Supercomputing
https://doi.org/10.1007/s11227-019-03018-x

An optimisation tool for robust community detection algorithms using content and topology information

Amhmed Bhih[1] · **Princy Johnson**[1] · **Martin Randles**[1]

Abstract
With the recent prevalence of information networks, the topic of community detection has gained much interest among researchers. In real-world networks, node attribute (content information) is also available in addition to topology information. However, the collected topology information for networks is usually noisy when there are missing edges. Furthermore, the existing community detection methods generally focus on topology information and largely ignore the content information. This makes the task of community detection for incomplete networks very challenging. A new method is proposed that seeks to address this issue and help improve the performance of the existing community detection algorithms by considering both sources of information, i.e. topology and content. Empirical results demonstrate that our proposed method is robust and can detect more meaningful community structures within networks having incomplete information, than the conventional methods that consider only topology information.

Keywords Social networks · Community detection · Hybrid similarity · Incomplete information networks

1 Introduction

Real-world networks are not random networks, and they usually exhibit inhomogeneity and reveal a high level of order and organisation [1]. An interesting feature that real-world networks usually present is the community structure property, under

✉ Amhmed Bhih
amhmed_bhih@yahoo.com; a.a.bhih@2011.ljmu.ac.uk

Princy Johnson
P.Johnson@ljmu.ac.uk

Martin Randles
M.J.Randles@ljmu.ac.uk

[1] Department of Electronics and Electrical Engineering/Computer Science, LJMU, Liverpool L3 3AF, UK

Published online: 19 October 2019

which the topology of network is organised into modules commonly called communities or clusters [2].

In many real-world network structures such as social networks and the World Wide Web, in addition to the link information, nodes also have their attribute values referred to as attribute/content information. For example, in a social network, the nodes' properties could describe the roles of a person while the topological structure represents relationships among a group of people.

Most of the existing approaches found in the literature make use of either link information or attribute information analysis alone for community detection. However, in real-world networks, neither piece of information on its own is sufficient in determining good clusters of the network. The link information is usually sparse and noisy. On the other hand, relying on the attribute information alone could mislead the process of community detection. For example, the process may not identify the strength of a node's relationship with its neighbours correctly. Consequently, by taking into account only one source of information, the algorithm may fail to detect accurately the entire community memberships. Considering more than one source of information for community detection could produce meaningful clusters and improve the robustness of the network. For instance, when considering both the attribute information and connectivity information, if either one source of information is noisy or missing, the other could make up for it. Therefore, the proposed approach will consider attribute information and structure information. The structure information consists of shared neighbours information and connectivity information aspects of the network [3].

1.1 Research benefit and its impact

Community structure is a common and important topological characteristic of many real-world complex networks. Nodes belonging to a tight-knit community are more than likely to have other properties in common [4]. The determination of communities in the networks can provide powerful insights into the structure of networks, help to better understand the structural make-up of the networks and analyse complex phenomena at different scales [5, 6]. Thus, the outcome of this research work has valuable applications in several fields such as biology, social science, physics, computer science and business science [5, 7].

In social networks, for example, analysis of community detection is extremely useful in the context of many applications, including customer segmentation, vertex labelling, recommendations and link inference [8]. Community structure is important not only in social networks, but also in various other networks. For example, determination of community structure in the Internet can address questions such as how to route data as packets in an efficient way, how to reduce the time consumption for such traffic and what is the fast and safe path to consider to reach the destination. It can go further in depth, by elucidating questions like how computer viruses are spreading through the Internet and what mechanisms they follow to hit organisations. Also in dark networks, community structure can reveal the hidden relationships between individual terrorists [9]. Similarly, in the case of the World Wide Web

(WWW) pages related to the same subject are typically organised into communities, so that the identification of these communities can help the task of seeking for identifying the category of the network as well as understanding its dynamic evolution and organisation [10].

Clustering is an important technique in mobile ad hoc and sensor networks [11] for the improvement of certain management, e.g. energy consumption and communication tasks. Yu and Chong [12] reported that the cluster structure is an effective topology that could provide many benefits in the context of wireless sensor networks (WSNs). It could be used to increase the system capacity by spatial reuse of resources. Furthermore, it improves routing performance, because of the fact that the set of cluster heads and cluster gateways can normally form a virtual backbone for inter-cluster routing, and thus, the generation and spreading of routing information can be restricted to this set of nodes. Additionally, they stated that the cluster structure makes an ad hoc network appear smaller and more stable in the view of each mobile terminal.

1.2 Related work and scope of study

1.2.1 Related work

Community detection is an active area of network science research and over the years, a wide variety of community detection algorithms have been proposed to find the communities in the network. Community detection is also named as graph partitioning, in much of the literature [13, 14]. It is tempting to suggest that community detection and graph partitioning are really addressing the same question, since both their aim is to identify groups of nodes on a network that are better connected to each other than to the rest of the network. However, it is very important to stress that the task of graph partitioning and community detection can be distinguished from one another based on whether the experimenter fixes the number and size of the groups or it is unspecified [15]. Graph partitioning is the problem of partitioning a graph into a predefined number and size of clusters. It has been pursued particularly in computer science and related fields with applications in parallel computing and very-large-scale integration (VLSI) design, whereas, in the community detection, which has been pursued by sociologists and more recently by physicists and applied mathematicians, with applications especially to social and biological networks, the number and size of clusters are unspecified. Furthermore, the goal in the former is usually to identify the best division of a network regardless of whether or not a good division existed. In case there are no good divisions existing, the least bad one will be identified as the solution. On the other hand, in the latter, the algorithm only divides the network when good divisions exist and leave the network undivided in case there are no good divisions existing [3, 15].

The community detection algorithms can be classified in different ways, and depending on the selected criteria, one algorithm can belong to more than one category. Among them, those based on modularity maximisation form the most

prominent family of community detection algorithms such as fastgreedy algorithm [16] and Louvain algorithm [17].

Fastgreedy algorithm is an agglomerative hierarchical clustering method proposed by Newman [16]. The algorithm greedily maximises the modularity function Q and starts the process by assigning a different community to each node in the network. Then, at each stage in the process, the pair of clusters that yields greatest increase of modularity or smallest decrease is merged until only one cluster remains containing all nodes in the network. The whole procedure can be represented by a dendrogram (hierarchical tree) that illustrates the order of the mergers. Cuts through the dendrogram at different levels give different partitions into communities. The optimal community cluster can be found by cutting the dendrogram at the level of maximum Q.

Louvain algorithm is a hierarchical agglomerative optimisation method proposed by Blondel et al. [17] and attempts to optimise the modularity of a partition of the network. The optimisation is performed in two steps that are repeated iteratively. This algorithm starts with each node in the network belonging to its own community. Then, in the first step and for each node in the network, the algorithm uses the local moving heuristic to obtain an improved community structure by moving each node from its own community to its neighbours' community and evaluating the gain of modularity associated with the moving of the node. The node is then placed in the community for which the modularity change is the most positive. If none of these modularity changes is positive, the node stays in its original community. This process is applied repeatedly and sequentially for each node until all the nodes in the network are considered, and no further improvement can be achieved. This concludes the first step. The second step of the algorithm consists of building a new network from the communities discovered in the first step. Therefore, the individual nodes in the new network are the individual communities from the first step. In this new network, there will be an edge between two nodes if there were edges between the corresponding two communities in the previous step. The weights of those new edges are the sum of the weights of the edges between nodes in the corresponding two communities. The edges between nodes of the same community in the first step will lead to self-loops for this community node in the new network. Once the second step is completed, it is possible to replay the first step and iterate again if necessary. The two steps repeat iteratively and stop when there is no more change in the modularity gain, and consequently, a maximum modularity is obtained.

Another popular method widely used to find communities in the network is based on the random walk. An example includes Walktrap (WT) algorithm which is proposed by Pons and Latapy [18]. Walktrap algorithm is based on the principle that random walks on a network tend to get 'trapped' into densely connected parts defining the communities. In this method, the authors propose using a node similarity measure based on short walks to capture structural similarities between nodes instead of modularity to identify community via hierarchical agglomeration. The algorithm starts by assigning each node to its own community, and the distance for every pair of communities is computed. Communities are merged according to the minimum of their distances and the process iterated. After $n-1$ steps, the algorithm

finishes and gives a hierarchical structure of communities called a dendrogram. The best partition is then considered to be the one that maximises modularity.

Information theoretic algorithms are another major type of community detection clustering algorithms that use the concept of information theory to find community clusters in the network. Infomap algorithm is an example of information theoretic algorithms proposed by Rosvall and Bergstrom in [19].

Infomap algorithm characterises the problem of finding the optimal community clustering in the network as the problem of finding the most compressed (shortest) description length of the random walks on the network. It uses a random walk as a proxy for information flow in a network and minimises a map equation, which measures the description length of a random walker, over all the network clusters to reveal its community structure. To represent the community structure, the algorithm uses a two-level nomenclature based on Huffman coding: a level to distinguish communities in the network and the other to distinguish nodes in the community. In practice, the random walker is likely to stay longer inside communities, and therefore, in the process of finding a community containing few inter-community links, only the second level is needed to describe its path, leading to a compact representation.

However, most of these algorithms are classified as global algorithms, which require access to the information of the entire network and make use topology information and largely ignore the attribute information [2].

1.2.2 Background and scope of study

Another property of similar interest is transitivity or global coefficient clustering, which is defined as the tendency between two nodes to be connected if they share a mutual neighbour [20]. In terms of network topology, transitivity is defined as the presence of a heightened number of sets of three vertices with edges between each pair of nodes (triangles) in the network.

Empirical studies have found that the concept of transitivity applies in about 70–80% of all cases across a variety of small group situations [21, 22]. Huijuan and Shixuan [23] proposed a graph clustering algorithm called SNGC that considers both connectivity between nodes and shared neighbours. Their experimental results show that the proposed algorithm provides promising results and could be applied to the analysis of social networks, computer networks, bioinformatics, etc.

Another common occurrence in networks is that similar nodes associate with each other more often than others (e.g. in social networks, people choose to be friends with people who share their beliefs). This property is known as homophily [24]. Traud and Kelsic [25] show that a set of nodes' attributes can act as the primary organising principle of the communities. Several studies have been performed to investigate this phenomenon of homophily, which is summarised in McPherson et al. [24].

There have been modifications and revisions to many methods and algorithms already proposed. A comprehensive survey of community detection in graphs has been done by Fortunato in [2]. Other reviews available in the literature are by Bedi and Sharma in [26] and Plantié and Crampes in [27].

Recently, there have been several studies [28–33], [34] showing that the combination of attribute and link information to detect communities in a network can improve the clustering quality. Most of these studies propose new algorithms that aim to use both sources of information; however, most methods use all attributes the same way without considering which ones may influence the community structure more, and lack the flexibility of balancing the information coming from network adjacency matrix (link information) and its node attributes.

Considering more than one source of information for community detection could produce meaningful clusters and improve the robustness of the network. Therefore, a pre-processing approach that considers both the attribute information and connectivity information aspects of the network for community detection is presented in this work. It should be noted that this work does not attempt to introduce a new community detection algorithm and rather proposes a pre-processing step to improve the performance of the existing community detection algorithms and enable them to execute in unreliable data network environments with better results.

In this paper, a network is represented as an undirected network $G=(V, E, A)$, where V is the set of nodes and E is set of edges between nodes. Each node $Vi \in V$ is associated with an attribute vector $(\mathrm{Att}_i^1, \ldots \mathrm{Att}_i^d)$, where d is the attribute dimension and i represents the node ID.

The main goal of this work is to find K non-overlapping communities in the network where the community (C) is defined as a list of non-empty node subsets: $C=\{C_1, C_2, \ldots, C_k\}$, and $V = \cup_{i=1}^{k} C_i$ that satisfy $C_i \cap C_j = \varnothing$ for any $i \neq j$.

1.3 Contributions arising from this work

During the past decade, the problem of community detection in networks has drawn a great deal of attention and several algorithms have been proposed. Recently, several studies have proposed methods that make use of both attribute and link information to detect communities in a network. However, as mentioned in the previous section, most of these studies propose new algorithms that aim to use both sources of information, use all attributes the same way without considering which ones may influence the community structure more, and lack the flexibility of balancing the information coming from network adjacency matrix and its node attributes. Additionally, none of the studies examines the quality and the number of community structures that could be identified in the network when some of the links are missing, i.e. noisy network environment.

The aim of this work is to design and implement a method that seeks to improve the performance of the existing community detection algorithms for incomplete networks. Hence, to the best of our knowledge, this is the first study on the community structure that seeks to:

1. Design and implement a unique pre-processing approach for the state-of-the-art community detection algorithms by tightly integrating the attribute information, shared neighbours and connectivity information aspects of the network to produce a new matrix.

2. Study the correlation between communities and attributes in the network and introduce weight detection attribute model to learn the degree of contributions of different attributes based on the impact of attribute on the community structure.
3. Evaluate the performance of pre-processing approach within incomplete, networks.

1.4 Structure of the paper

This paper is organised as follows: the experimental datasets along with the quality metrics for assessing the network clustering results are discussed in Sect. 2. Section 3 investigates the correlations between attributes and community structure of the network. Section 4 describes the novel proposed method along with a similarity matrix, used to weight the links between nodes in the network. Section 5 briefly presents the experimentations and evaluates the results of the proposed approach against the benchmark algorithms. The conclusion and future work are presented in Sect. 6.

2 Datasets and performance metric

2.1 Datasets

In order to investigate the correlations between attributes and community structure and to evaluate the proposed approach, anonymised Facebook datasets as introduced by Traud et al. [35] and [25] are used. The Facebook datasets are undirected and unweighted. The datasets were recorded on a particular day in September 2005 and contain Facebook networks from 100 different American university networks whose nodes represent users and whose links represent friendships between users. Attribute information about each user is also provided. Each user has seven node attributes: a student/faculty status flag, gender, major, second major/minor (if applicable), dormitory (house), year and high school. In this work, four networks from 100 Facebook datasets are used. In particular, the Caltech36, Reed98, Haverford76 and Vassar85 datasets, which contain 769, 962, 1446 and 3068 nodes and 16,656, 18,812, 59,589 and 119,161 edges, respectively, are used.

For more information about dataset, interested readers may refer to work by Traud et al. in [35] and [25]. However, the proposed approach in this work is not limited to the social networks but can be applied to many kinds of graph structures.

2.2 Performance metrics

To quantify the performance of the proposed approach, the quality of the obtained community structures is evaluated based on the modularity, number and size of detected communities.

Definition 1 modularity (*Q*) Modularity (Q) is a prominent measure for the quality of a community structure introduced by Newman and Girvan in [36], and it has become a widely accepted quality of measure for community detection. Modularity states that a good cluster should have a bigger than expected number of connections between the nodes within modules and a smaller than expected number of connections between nodes in different modules. The higher the value of modularity, the better its community strength.

Formally, modularity can be defined as [2]:

$$Q = \frac{1}{2|m|} \sum_{ij} \left[A_{ij} - \frac{K_i K_j}{2|m|} \right] \delta_{c_i c_j} \tag{1}$$

where A_{ij} is an element of the adjacency matrix, K_i is the degree of node i. $\delta_{c_i c_j}$ is the Kronecker delta symbol, which is equal to 1 if $c_i = c_j$ and 0 otherwise, and c_i is the label of the community to which node i is assigned.

3 Correlation analysis

3.1 Shared neighbours

In order to measure how likely any two nodes with a common neighbour are themselves connected, the clustering coefficient of each node in the network is calculated.

Definition 2 clustering coefficient CCO The node clustering coefficient C_i of a node i is defined as the ratio of the number of edges connecting the neighbours of i to the total possible number of such edges of i, and K_i is the degree of node i [10]:

$$\mathrm{CCO}_i = \frac{2L_i}{K_i \left[K_i - 1 \right]} \tag{2}$$

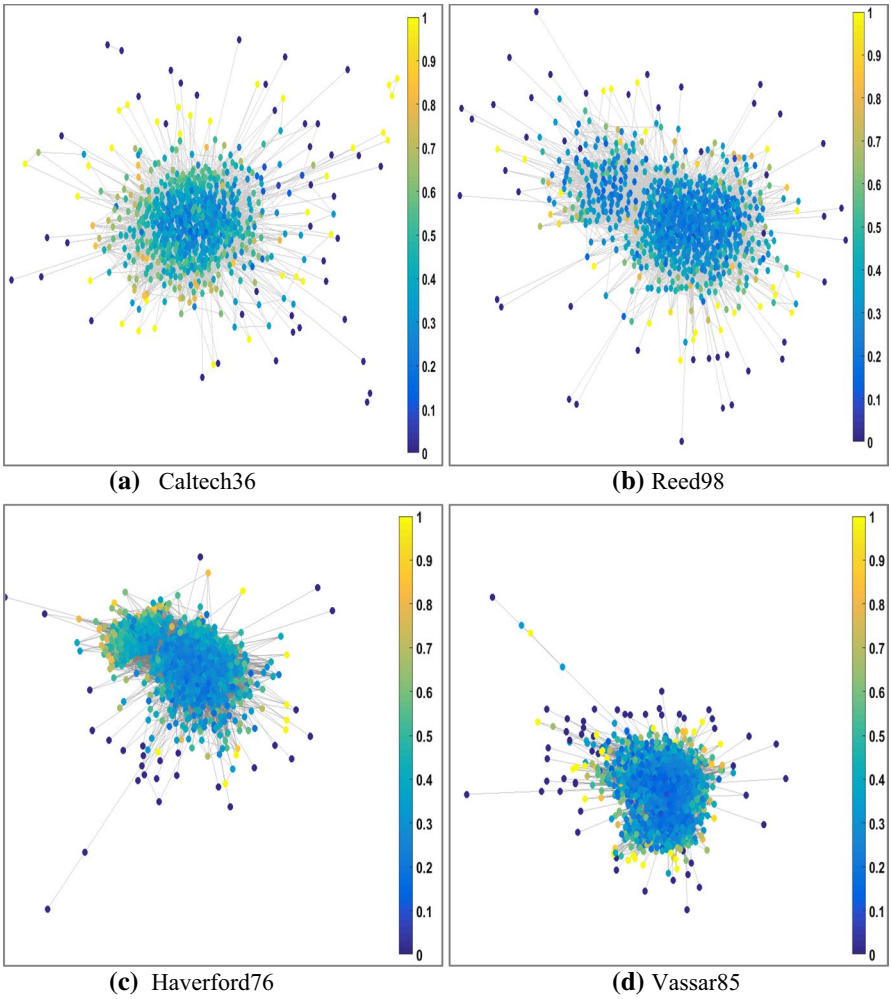
where L_i is the number of edges between neighbours of node i.

The clustering coefficient for the whole network is the average of the local values C_i:

$$\mathrm{CCO} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{CCO}_i \tag{3}$$

where n is the number of nodes in the network [10].

Figure 1 shows the visualisation results of the cluster coefficient for each node in the four datasets. In this figure, colours of nodes correspond to values of their corresponding clustering coefficients. As can be seen, there are some nodes that have high clustering coefficients, which indicates strong connectivity between each other. In other words, they are more prone to be in the same cluster. Furthermore,



(a) Caltech36 **(b)** Reed98

(c) Haverford76 **(d)** Vassar85

Fig. 1 Visualisation results of node clustering coefficient for subset of four datasets (colour figure online)

the clustering coefficient for the considered networks is 0.4288, 0.3304, 0.3268 and 0.2487 for Caltech36, Reed98, Haverford76 and Vassar85 datasets, respectively.

It is clear from the above discussion that the shared neighbours' information can be used to describe the nature of connections between nodes in the network. This should motivate the use of shared neighbours' information in detecting community clusters in the network.

3.2 Correlation of communities and attributes

For the sake of computing the correlation between connectivity of nodes and their attributes, the nodes are clustered based on their attributes in which the nodes whose attributes are similar are grouped together to form a cluster. Also, four different

community clustering algorithms, which are Fast Modularity [37], Louvain [17], leading eigenvector algorithm [38], and Walktrap [39] are applied on the datasets to find the communities. Then, the correlations between the resulting communities from these algorithms and the attributes are measured using Jaccard similarity index, which was introduced by the Paul Jaccard in [40].

Figure 2 presents the Jaccard similarity index for four different community detection algorithms with each attribute over the four networks in the Facebook dataset. It is interesting to notice that for the same dataset, the order of the correlation strength across different attributes is not same and varies from one community clustering algorithm to another. For example, in Reed98 dataset, if the agreement with the Fast Modularity algorithm is considered, the most agreement is observed with the attribute 'student faculty'. On the other hand, Louvain algorithm performs the best if the agreement with the 'year' is considered. This is due to the fact that each algorithm differs on how they treat the nodes and assign them to different communities with different size and number of communities.

Even though there exists a difference in attribute ranking across different algorithms and datasets, as an overview, the most agreements are observed with student faculty, gender, year and dormitory attributes. However, in computing the correlation between attributes and community structure, Traud and Kelsic [25] reported that the order of correlation strength is significantly dependent on the agreement index used and not consistent across different indices.

Observing the correlation between the attributes and the communities in the network indicates that the attribute information is a source of data that can be used to perform the community clustering task. Furthermore, based on the homophily property of a network as shown above it is clear that the linked nodes are more likely to share similar attributes. However, the attributes do not have the same influence as the community structure and some attributes weigh more than others in their influence. Thus, the impact of different attributes on communities needs to be known and properly weighted according to their influence on the community structure. This will balance the role of network information and node attributes.

4 The proposed optimisation approach

The proposed approach could be defined as a pre-processing phase for conventional community clustering algorithms, which takes a graph $G=(V, E, A)$, the weight of attributes (W) and two more weighting factors (α and β) as inputs. α is used to weight the contribution between connectivity information and both attribute and shared neighbours' information. β is used to weight attribute information to the number of common neighbours. However, these weighting factors (W, α, β) can be either provided as part of the input if they are known a priori or calculated from the dataset.

The proposed approach returns a hybrid similarity matrix. The hybrid similarity matrix is a weighted combination of attribute information, shared neighbours' information and connectivity information between the nodes. Once the proposed

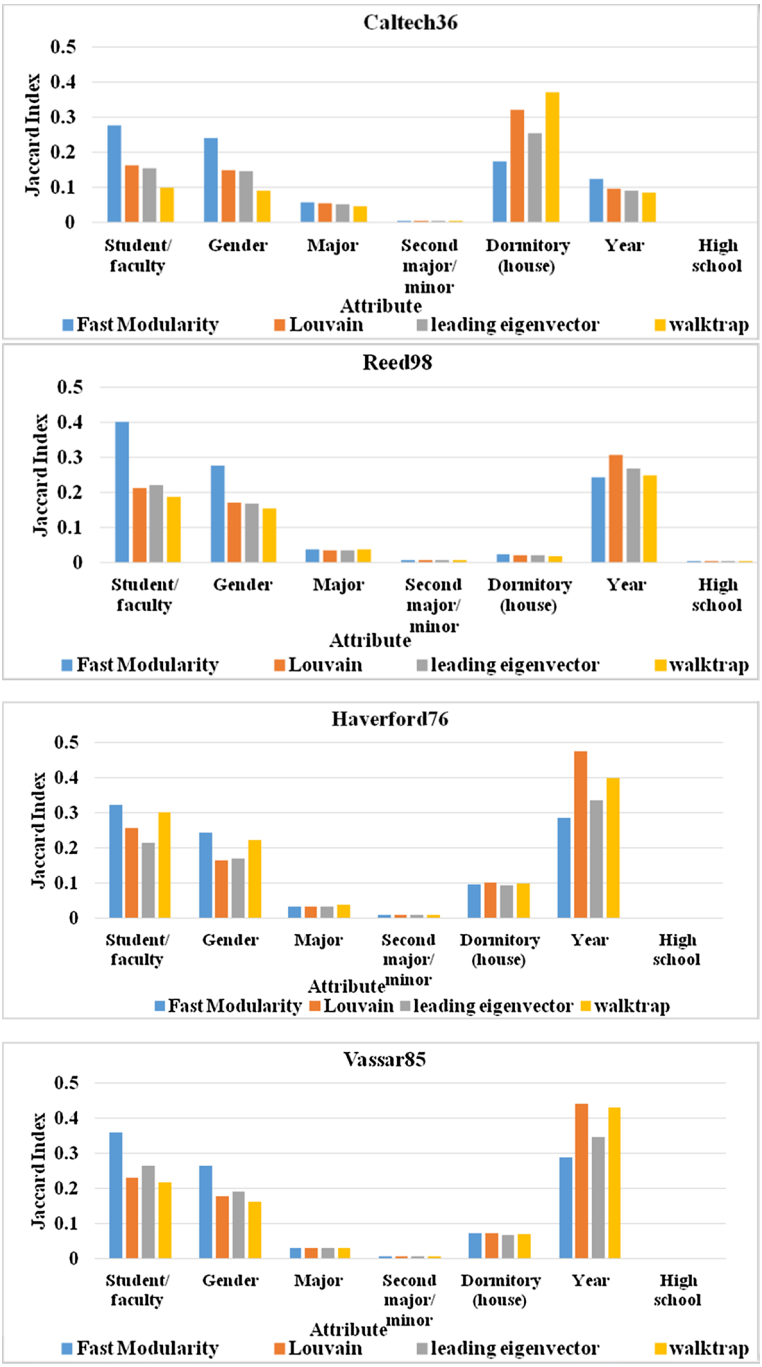


Fig. 2 Agreement of different community detection algorithms with each attribute, for a subset of four datasets

approach constructs the hybrid similarity matrix, it can be integrated with any of the state-of-the-art clustering algorithms proposed for weighted graph (e.g. Newman fast greedy algorithm, Louvain algorithm, Newman algorithm based on leading eigenvector of a modularity matrix or Walktrap algorithm) to extract optimum community clusters.

4.1 General architecture

The general architecture of the proposed approach is shown in Fig. 3. As can be seen in the figure, the approach has two phases, namely the parameter learning phase and information aggregation phase. The aim of the first phase is to extract optimal parameters, whereas the second one is used to build a hybrid similarity matrix.

We formally describe the generative process of hybrid similarity matrix as the following:

$$H_{\text{sim}}(i,j) = \propto A(i,j) + (1- \propto)\left[\beta Wa_{\text{sim}}(i,j) + (1-\beta)SN_{\text{sim}}(i,j)\right] \tag{4}$$

$$Wa_{\text{sim}}(i,j) = WA_{\text{sim}}(i,j) \tag{5}$$

where $H_{\text{sim}}(i,j)$: hybrid similarity matrix, A: adjacency matrix (matrix representation of exactly which nodes in the network contain edges between them), $Wa_{\text{sim}}(i,j)$: the weighted attribute similarity between a pair of nodes (i, j), α: the weighting factor used for the contribution of connectivity information to the attribute information and shared neighbours information, β: the weighting factor used for the contribution of attribute information to the number of common neighbours information, $SN_{\text{sim}}(i,j)$: shared neighbours similarity between nodes i and j, $A_{\text{sim}}(i,j)$: the attribute similarity between a pair of nodes (i, j) in network $G=(V, E, A)$, and W: a matrix containing the weights of each attribute of the node in the network.

Definition 3 shared neighbours Given a graph $G=(V, E)$, for a node $i \in V$, the neighbours of node i are nodes that directly connect to node i and is denoted by $\Gamma(i)$.

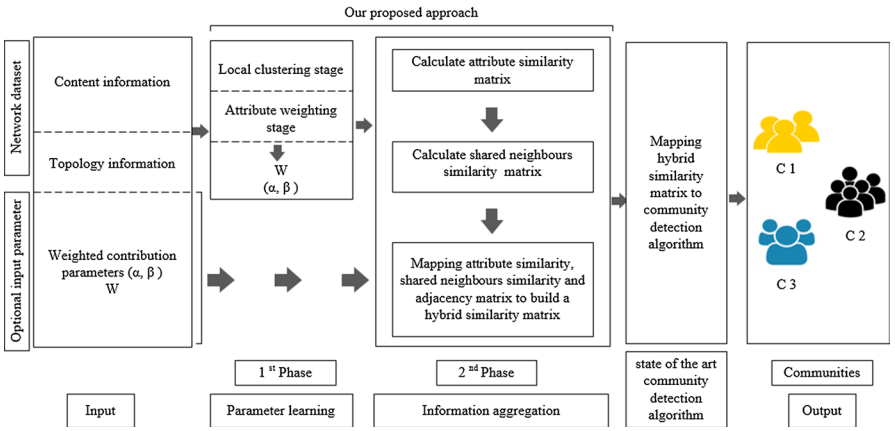


Fig. 3 System architecture for the proposed approach

The shared neighbours of node i and j are the nodes that both directly connect to nodes i and j. It is defined as:

$$SN(i,j) = \{\Gamma(\mathrm{i}) \cap \Gamma(\mathrm{j})\}. \tag{6}$$

The shared neighbours similarity between nodes i and j is calculated by dividing the number of shared neighbours between them by the maximum degree of i and j nodes. It is defined as:

$$SN_{\mathrm{sim}}(i,j) = \frac{SN(i,j)}{\max\left[K_i, K_j\right]} \tag{7}$$

where $SN(i,j)$: shared neighbours between nodes i and j and K_i: degree of node i

In the hybrid similarity matrix, as is defined in Eq. 4, the strength of relationship between nodes is determined by attribute information, connectivity information and shared neighbours and controlled by two weighting parameters (α and β). The α and β weighting parameters can be given as part of the input values by the human agent based on their knowledge of the data structure and their perception of the importance of each attribute. However, choosing the right weighting values of attributes without a priori knowledge of the network is a challenging task. Hence, the values of the attribute weighting factors (W) in the proposed approach need to be set carefully. In the following sections, the two phases of the proposed approach (the parameter learning phase and information aggregation phase) will be discussed in detail to provide guidelines on how to set these parameters.

4.2 The parameter learning phase

Since the goal of utilising details on attribute information, shared neighbours and connectivity information in this work is to get the best community clusters for the network, the attributes of the nodes should be weighted in such a way that greater weight is given to the more influential attributes and smaller weights for the less influential. Determining the influence and thus the weights of the attributes correctly will enhance the community structure algorithm and improve the detection of communities in the networks. The main purpose of the proposed attribute weighting technique is to search for small groups of nodes (initial clusters) that contain more internal connections (links between nodes in the group) than external connections (between nodes of the group and nodes in other groups) and then find the attribute similarity between nodes in the same groups to get the influence factor for each attribute.

To accomplish this, the parameter learning phase, as shown in Fig. 3, is subdivided into two stages: local clustering stage and attribute weighting stage. Local clustering phase is to extract dense nodes from the network to form the initial clusters. These initial clusters are local small ones, far from being the optimal result, and are only used in the second stage to weight the attributes of each node in the network as well as estimate the α and β parameter values.

In the local clustering phase, the initial clusters are obtained by applying the first phase of the DICCA algorithm proposed by the authors in [41], named local clustering phase. The basic idea of the local clustering phase in DICCA consists of picking up m nodes to be originators in which the m nodes are spread out across the entire region of the network and assigning each node to the closest originator to form a cluster.

The attribute weighting stage is then applied to find the strength of the weighting for each attribute based on the structures of current clustering results. During the attribute weighting stage, the set of attributes for each node are weighted according to its influence in the community in which the highly influential attributes are assigned with high strength weights; meanwhile, the less influential attributes are assigned with low strength weights.

In order to find the attribute weighting, it is necessary to measure the proximity between pairs of nodes in the initial clusters based on their attributes. To do so, the attribute similarity metric needs to be defined first.

4.2.1 Attribute similarity metric

The attribute similarity between nodes V_i and V_j within the same cluster is determined by examining each of d set of attributes on the two nodes and reflect on the strength of the relationship between them in terms of their attribute values.

It must be emphasised that irrespective of the similarity metric considered to find the weight of attributes, first, the similarity between the attribute values of each pair of nodes belonging to the same local cluster needs to be calculated. The procedure is as follows:

Let $X^i_{N.d}$ be the similarity matrix for cluster i with N nodes each with d attributes, the local attribute weight for cluster i is obtained by adding the appropriate dimension attribute of each node in the cluster to form a vector of $1 \times d$ size and expressed as:

$$\mathrm{LW}^i_d = \frac{1}{N} \sum_{i=1}^{d} \left(X^i_{N.d} \right). \tag{8}$$

The weighting for the entire network is then calculated by adding the corresponding attribute of each local attribute weight (sum of the vectors) to form another vector in $1 \times d$ size. It is formally defined as:

$$W = \frac{1}{m} \left(\sum_{i=1}^{m} \mathrm{LW}^i_d \right) \tag{9}$$

where LW^i_d: the local attribute weight for cluster i and W: attribute weights of the node in the network.

It is worth mentioning that the weights assigned to the attributes in the parameter learning phase $\mathrm{LW} = \{Lw_1, Lw_2 \ldots Lw_m\}$ range between 0 and 1.

Whether or not a certain subset is optimal depends on the similarity metric employed. The question about what are the best similarity measures between nodes to choose, for different types of attribute data, is beyond the scope of this work. In this work, a Jaccard similarity coefficient is used to define the attribute similarity

between nodes in the same cluster and to find the weight of attributes (W) during the parameter learning phase. For an overview of the research work on determining the most meaningful similarity measures in various fields and for different types of data, see [42, 43].

Definition 4 Jaccard similarity Given a network $G=(V, E, A)$, for any pair of nodes V_i, $V_j \in V$, the Jaccard similarity between nodes V_i and V_j with respect to attribute is indicated as $J(A_i, A_j)$ and is defined as the size of the intersection divided by the size union of the data sets, as given below [44]:

$$J\left(A_i, A_j\right) = \frac{\left|A_i \cap A_j\right|}{\left|A_i \cup A_j\right|} \tag{10}$$

where $J(A_i, A_j)$ returns a value between 0 and 1, with 0 denoting no similarity and 1 denoting identical sets.

Furthermore, since in this work Jaccard similarity is used to measure attribute similarity between nodes, the $X^i_{N.d}$ could be defined as the Jaccard similarity matrix for cluster i. The weighted attribute similarity $Wa_{\text{sim}}(i,j)$, between any two nodes i and j is defined as follows:

$$Wa_{\text{sim}}(i,j) = \frac{\sum_{L=1}^{d}(W_L * \left[\text{Att\_}i_L \cap \text{Att\_}j_L\right])}{\sum_{L=1}^{d}(W_L * \left[\text{Att\_}i_L \cup \text{Att\_}j_L\right])} \tag{11}$$

where each node has d attributes and Att_i is the attribute vector of node i.

The pseudo-code outlining the entire procedure with Jaccard similarity is listed in Algorithm 1.

4.2.1.1 Effect of α and β on the quality of community structure When considering to select the values for the two weighting factors (α and β), the type of emphasis on one of the network parameters needs to be considered. For example, emphasis on the connectivity information source means that the parameter α should be greater than 0.5. On the other hand, emphasis on attribute and shared neighbours information means that α should be less than 0.5. The same argument holds good for the parameter β, i.e. β greater than 0.5 indicates that attribute node information source has more contribution than the information related to the number of common neighbours. In the networks, the weighted combination of attribute information, shared neighbours and connectivity information is not same and the values of α and β need to be selected carefully. However, in practice without any prior domain knowledge, it is quite difficult to scale the contribution of each source of information.

In order to determine the effects of varying α and β parameters on the quality of community clustering and thereby to determine the parameters' selection range, four different datasets are used to track how the community clustering changes when the

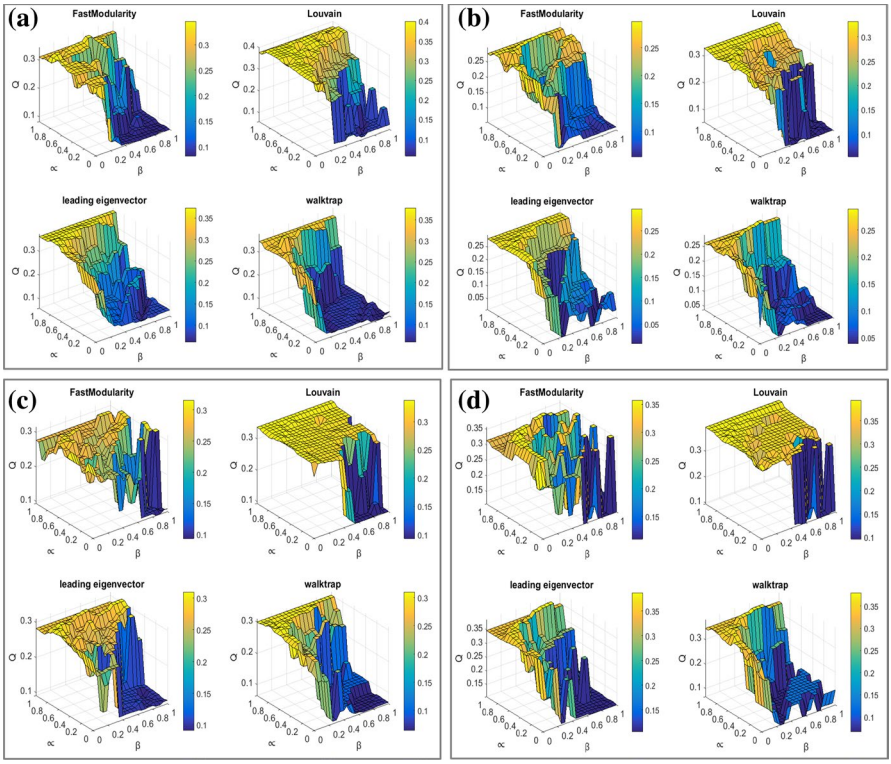


Fig. 4 **a**–**d** Modularity value achieved by four community clustering algorithm dataset using different value of α and β on: **a** Caltech36, **b** Reed98, **c** Haverford76, **d** Vassar85 datasets

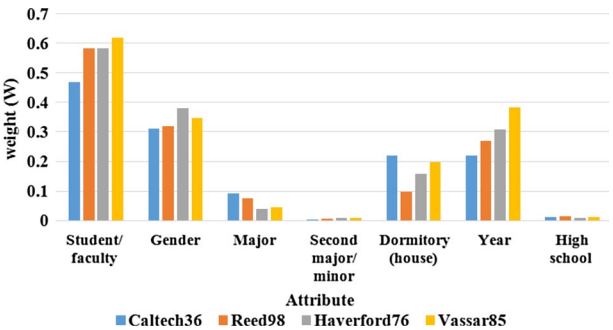


Fig. 5 Attribute weights for four datasets

values of α and β are varied from 0.1 to 1 with a step size of 0.1. Also, modularity index is used to evaluate the quality of community detection.

Figure 4 shows how the two parameters influence the community clustering quality. The X-axis and Y-axis in the figures represent the values of α and β, respectively, while the Z-axis represents the modularity score. As can be clearly seen from

Fig. 5a–d, the modularity is remarkably robust to the choice of parameter values. When $\alpha=\beta=0$, the modularity of community detection is ≥ 0.25 for most of the algorithms for all the datasets. However, it is worth mentioning that $\alpha=\beta=0$ indicates that the information used to find the community clustering is just based on the number of common neighbours $H_{sim}(i,j) = SN_{sim}(i,j)$.

As an overview, with an increasing value of β, the quality of community clustering decreases for a constant value of α. On the contrary, with an increasing value of α, the quality of community clustering increases slightly for a constant β value. It is also noticed that for values of $\alpha<0.6$ the modularity is dramatically affected by varying the value of β. The modularity fluctuates between 0.01 and 0.4, and it becomes relatively stable when α value ranges between 0.6 and 0.7. However, the modularity becomes almost stable for the vast majority of β values when $\alpha>0.7$.

Algorithm 1: The proposed approach

Input:

adj: adjacency matrix.
Att: An attribute nodes matrix.
Optional input parameter:

W: a matrix containing the weights of each attribute for each node in the network.
∝: The weighted Contribution of connectivity information to the attribute information //and shared neighbours information.
β: The weighted contribution of attribute information to the number of shared //neighbour information.

Output:

K: A set of communities in the network.

for each Node i ∈ adj
  $A_{sim}(i,j) = \sum_{L=1}^{d}[Att\_i_L \cap Att\_j_L]) / \sum_{L=1}^{d}[\ Att\_i_L \cup Att\_j_L]\ )$ //get attribute //similarity matrix between *i* &*j* where *i*≠*j*
  $\Gamma(i) \leftarrow$ get the neighbours of node (*i*)
  $k_i \leftarrow$ get the degree of node (*i*)
  end
$SN(i,j) = \{\Gamma(i) \cap \Gamma(j))\}$ //get the number of shared neighbours between each nodes
$SN_{sim}(i,j) = SN(i,j) / max[K_i, K_j]$ // shared neighbours similarity between nodes *i* // and j where *i*≠*j*
C= **local clustering phase** (*adj*) // run the first phase of DICCA algorithm
for each cluster $lc \in C$
  For each pair of nodes $i,j \in lc$
    $X_{N.d}^{lc} \leftarrow |Att\_i \cap Att\_j| / |Att\_i \cup Att\_j|$ // Jaccard similarity matrix for cluster lc
    end
  N← get number of nodes in *lc*
  $LW_d^{lc} = \frac{1}{N}\sum_{i=1}^{d}(X_{Nxd}^{i})$
  End
  m← get number of initial clusters in *c*
  if (*W* not provided as an input parameter**)**
    $W = \frac{1}{m}(\sum_{i=1}^{m} LW_d^i)$
    end
  if (∝ not provided as an input parameter**))**
    $\propto = avg(W)$
    end
  if (β not provided as an input parameter**))**
    β ***=0.5***
    end

$$Wa_{sim}(i,j) = \sum_{L=1}^{d}(W_L * [Att_{i_L} \cap Att_{j_L}]) \Big/ \sum_{L=1}^{d}(W_L * [\ Att_{i_L} \cup Att_{j_L}]\ )$$

$$H_{sim}(i,j) \leftarrow \propto . Adj(i,j) + (1-\propto)[\beta . Wa_{sim}(i,j) + (1-\beta) . SN_{sim}(i,j)]$$

$K \leftarrow$ **community *cluster*** $(H_{sim}(i,j))$
Return *K* return the final division of *adj*

Experimental results also demonstrate that the connectivity information is more useful than the shared neighbours' information and attribute information. Therefore, the value selected for α should be greater than or equal to 0.5. For the datasets considered in this work, high modularity values are obtained when $\alpha > 0.7$.

With regard to these two parameters α and β, there is no straightforward way to fit them to datasets and different datasets may require different parameter values. However, based on the above argument, in order to better exploit the sources of information and obtain optimum robustness in the detection of community clusters in the presence of noise, the value of α is set based on the weights of attributes (w) as follows:

$$\alpha = \text{avg}(w). \tag{12}$$

In this work, to avoid a cumbersome decision process, equal importance is given to shared neighbours and attribute information in which $\beta = 0.5$ is set in all the following performed experimentations.

4.3 Information aggregation phase

The information aggregation phase aims to build a weighted matrix, named hybrid matrix, based on the knowledge learned from the parameter learning phase. These weighted attributes w, α and β values are used to build a hybrid similarity matrix as defined in Eq. 4. In the hybrid matrix, the edges that link nodes do not have similar attributes or do not have shared neighbours will be punished and assigned with low strength weights, while the edges connecting similar nodes or having shared neighbours will be assigned with high strength weights. Also, there are some edges which will be added between the nodes to represent the attribute and shared neighbour similarity.

5 Experimentation and results

5.1 Experimental setup

In order to assess the effectiveness of the proposed approach to detect communities under an unreliable network structure, an experimentation has been conducted using four different Facebook dataset networks when some edges are missing, while the node attributes are fully available. Furthermore, for the sake of evaluation, edges are removed from the network at random and the number of removed links is increased from zero to half the number of edges in the network in steps of 5% of network edges.

In each experiment, the performance is computed using the results obtained by applying each of the four algorithms with and without applying the proposed approach as a pre-processing step. Each algorithm has been applied more than once on the data, and the experimental results presented are the average of ten simulation runs.

To quantify the performance of the proposed approach, the quality of the obtained community structures is evaluated based on the modularity, number and size of detected communities.

The performance of the proposed approach is evaluated in terms of repeatability and reproducibility when noise is introduced in the environment. This is measured by its ability to find the same ground truth communities detected under normal conditions even when noise is introduced. The outcome of the community clustering solution obtained by each algorithm with the original dataset (complete dataset) is used as a ground truth and compared against the outcome of the clustering solutions when a number of edges are progressively removed from zero to half the number of total edges in the network.

Moreover, for simplification, in the following sections when the proposed approach is combined with Fast Modularity algorithm (FA) it is referred to as Hybrid-FA; when combined with Louvain algorithm (LA) as Hybrid-LA; when combined with leading eigenvector (LE) as Hybrid-LE; and Hybrid-WA when combined with Walktrap algorithm (WA). Additionally, to facilitate comparison of results in line charts, the results obtained using the proposed approach are denoted by dashed line style with 'x' marker points.

It is worth mentioning that we have attempted to define and evaluate the computational complexity of this algorithm in [45]. While the exact mathematical model for the computational complexity of the pre-processing algorithm is harder to formalise, it could be represented using the computational model as $[\log (nm)^2]$, in which n is the total number of nodes in the network and m the number of edges.

5.2 Experimental results and discussion

In this subsection, the effectiveness and efficiency of the algorithm are assessed from two aspects. One is to evaluate the attribute weighted method proposed in this work along with the methodology used to set the parameter value. The other aspect is to integrate the proposed approach with well-known community clustering algorithms and make a comparison of the results achieved without the integration to show how the proposed approach can be used to improve the robustness and quality of those well-known community clustering algorithms.

5.2.1 Evaluation of attribute weighting method

As highlighted in Sect. 3, different attributes have different significance for assessing the similarity between the nodes in the same community clusters; therefore, the attribute weighting method is proposed. In this section, the performance of the proposed attribute weighting method is experimentally evaluated.

The evaluation is done by checking how well the weight of the attributes (W) obtained by the weighting method match with the actual important attributes and is presented in Fig. 2.

Figure 5 shows the attribute weights obtained by the weighting method for the four datasets under consideration. It is obvious that the attributes have different

weight strengths and order of importance for different datasets. However, looking at the attribute weights of the four data sets, it is clear that four specific attributes (student, gender, dormitory and year attribute) have the highest weighting values across all four data sets. Anyway, the remaining attributes (high school and major/minor attribute) do not have strong influence on the community structure, hence weighted with a very small value in the attribute weighting stage.

Moreover, the comparison between Figs. 2 and 5 shows that the parameter learning phase achieves almost the same results in most cases, whereas the attribute importance order is either same or only slightly different due to small differences in the attribute correlation. For example in Caltech36 dataset, the order of importance attributes are student, gender, year and house with attribute weight values 0.4695, 0.3102, 0.2195 and 0.2193, respectively. In comparison with Fig. 2 and for the case of the Fast Modularity algorithm as an example, the order is changed to student, gender, house and year attribute, achieving Jaccard index values of 0.2772, 0.2412, 0.1746 and 0.1239, respectively.

Furthermore, to evaluate the performance of the proposed weighting method in handling noisy data, Fig. 6 shows the values of attribute weight for the four largest weighted attributes obtained by the weighting method when the percentage of removed edges varied from 0 to 50%. From the figure, it is worth noting that the ordering of weights is remarkably stable and the attribute weighting method shows an effective performance by getting rid of the noisy datasets and correctly weights attributes according to their importance.

To further assess the parameters analysis phase, the number of initial clusters identified at local clustering stage along with the value of α against the per cent of removed edges, for the four datasets, is reported in Table 1.

The results in Table 1 indicate that the noise has no significant influence on the value of α. In other words, the method used to define α value (see Eq. 12) is somewhat stable. In addition, it is clear that local crusting tends to partition data to a larger number of initial clusters. Considering Reed98 dataset, for example, when the missing edges varied from 0 to 50%, the values of α and the number of obtained initial clusters were {0.808, 382} and {0.823, 446}, respectively.

It is also worth noting from Table 1 that the value of α is not related to the number of initial clusters found by the local clustering stage. In some cases, higher value of α is obtained when more initial clusters are found. For others, however, the value of α increases when fewer initial clusters are found. Considering Reed98 dataset, for instance, when the missing edges increased from 15 to 20%, both α value and the number of initial clusters increased from {0.814, 399} to {0.816, 405}, respectively. On the other hand and for the same dataset, when the missing edges increased from 5 to 10%, the value of α increased from 0.812 to 0.813; meanwhile, the number of initial clusters decreased by 3. However, the value of α for the four considered datasets is always higher than 0.75. This value is in agreement with what was observed in Sect. 4.1.1.1, where the connectivity information contains more useful information than the shared neighbours or attribute information ($\alpha \geq 0.5$) and to get high modularity the value of α should be higher than 0.7.

Overall, the results clearly demonstrate that the parameter learning method has the ability to extract essential and informative attributes and to weight them to reflect the relative importance of attribute in community clustering tasks.

5.2.2 Model performance

In this subsection, using the optimal parameters determined using the parameter learning phase (as discussed in Sect. 4.1), the performance of the pre-processing approach is evaluated.

5.2.2.1 Number of community clusters Since the number of communities in the networks is unspecified, the algorithms try to automatically detect the most appropriate number of communities by maximising the modularity. The variation in number of community clusters when different numbers of edges are removed is shown in Fig. 7. It is observed that the conventional algorithms are adversely affected by noise, so fail

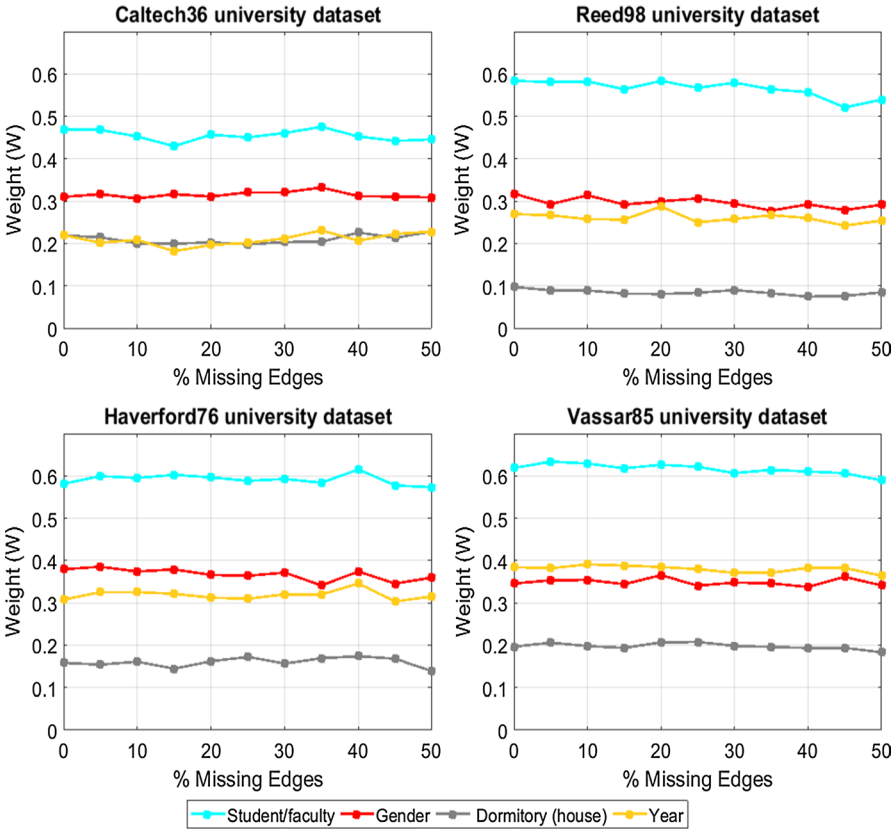


Fig. 6 Robustness of weighting method to the edge removal

Table 1 Results for four datasets

Dataset	Caltech36		Reed98		Haverford76		Vassar85	
%Missing edges	Number of initial clusters	α	Number of initial clusters	α	Number of initial clusters	α	Number of initial clusters	α
0	384	0.813	382	0.808	412	0.779	824	0.767
5	381	0.816	392	0.812	427	0.781	835	0.767
10	392	0.818	389	0.813	436	0.782	844	0.768
15	388	0.816	399	0.814	419	0.782	873	0.769
20	392	0.816	405	0.816	443	0.783	898	0.771
25	391	0.816	397	0.815	463	0.783	921	0.771
30	390	0.816	409	0.817	467	0.784	927	0.772
35	394	0.817	402	0.818	476	0.783	948	0.773
40	398	0.815	418	0.819	489	0.786	953	0.774
45	390	0.817	432	0.824	487	0.788	1003	0.776
50	387	0.811	446	0.823	514	0.788	1036	0.778

to account for appropriate community structures. Moreover, most cases result in an increasing number of communities with an increasing 5% of missing edges. The only exception is the LEA algorithm, which results in almost the same number of communities even without applying the pre-processing approach.

Considering Caltech36 dataset, for example, increasing proportions of edges are randomly removed from the network (from 0 to 50%), the number of communities detected by all conventional algorithms is changed from {10,10,12,72} to {39,39,10,104} for {FA, LA, LEA, WA} algorithms, respectively. Such behaviour can be explained by the fact that the conventional algorithms consider only topology information. On the other hand, the proposed approach considers attribute, shared neighbours and connectivity information. Since the nodes in the same community usually are not just highly connected but also have similar attributes and transitivity coefficient, the proposed approach uses attribute information to make up for the missing link information and to identify the community membership. Consequently, integrating the proposed approach with a conventional algorithm is more advantageous for discovering the most appropriate number of community structures than using the conventional algorithm on its own.

Walktrap algorithm when run on the dataset on its own failed to detect the appropriate number of communities, and compared to the other algorithms, the number of communities returned by Walktrap is extremely high for all considered datasets. However, applying the proposed approach as a pre-processing step to build the hybrid similarity matrix before applying the Walktrap community detection algorithm has significantly improved the performance to obtain just 8 clusters.

Furthermore, when the percentage of removed edges is increased from 0 to 50%, the number of clusters formed using the proposed approach is more similar to the original partition network when there is no noise applied. For example, in the case of Caltech36 dataset when 50% of edges are missing, the number of obtained

communities is {4,8,8,4} for {Hybrid-FA, Hybrid-LA, Hybrid LEA, Hybrid-WA} algorithms, respectively. This demonstrates that the proposed approach has the capability to extract relevant information from highly noisy datasets and make these algorithms quite robust to edge removal.

5.2.2.2 Size of community clusters To take a closer look at the sensitivity of the obtained communities to the noise, the average size of the obtained communities, when percentage of removed edges is increased from 0 to 50%, is investigated and shown in Fig. 8.

Considering Vassar85 dataset, for example, increasing the proportion of edges that are randomly removed from network (from 0 to 50%), the average community size detected by all conventional algorithms dropped from {614, 511, 438, 51} to {94, 95, 583,28} for {FA, LA, LEA, WA} algorithms, respectively. In contrast, combining the proposed pre-processing approach with the community clustering algorithms considered in this work results in community clusters with almost constant average size. This effect comes from the fact that since the conventional community identification is based only on the adjacency matrix, the number of community clusters obtained is heavily dependent on the number of links in the network, so as the percentage of missing edges increases, the clustering algorithm becomes less stable and the clusters become smaller. In contrast, this is not the case for the hybrid similarity matrix, which is based on different considerations (attribute information, shared neighbours information and connectivity between nodes in the network).

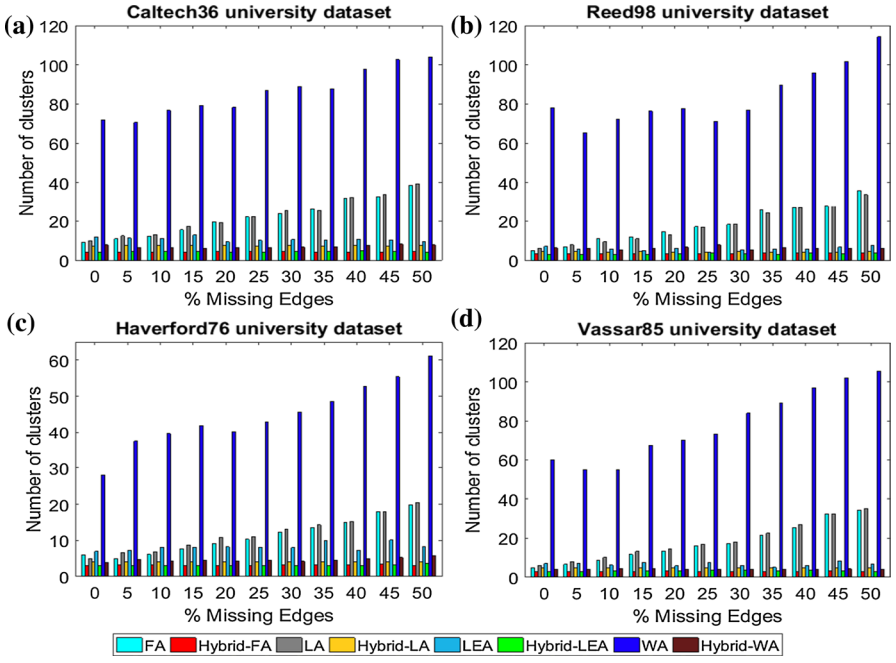


Fig. 7 Number of community clusters for: **a** Caltech36 university dataset, **b** Reed98 university dataset, **c** Haverford76 university dataset, **d** Vassar85 dataset

5.2.2.3 Modularity Regarding the quality of community clusters, the modularity metric is used as a scoring function to assess the quality of detected community clusters with and without applying the proposed pre-processing phase. Figure 9 shows the averaged Q values, plotted for each community detection algorithm. As shown in this figure, in most cases using the proposed pre-processing approach has resulted in a slightly lower modularity than the conventional community detection methods. However, the difference is negligible and the results suggest that the proposed approach is a promising and powerful tool to assist in the fine tuning of different sources of information in community clustering area.

Moreover, the comparison between Figs. 7, 8 and 9 shows that while the approach achieves a good modularity quality that is comparable with the conventional methods, the approach is significantly more effective in terms of both number and size of communities detected where the network structure is found to have some unreliable or missing information.

Table 2 shows the overall performance results of the proposed method using different types of source information.

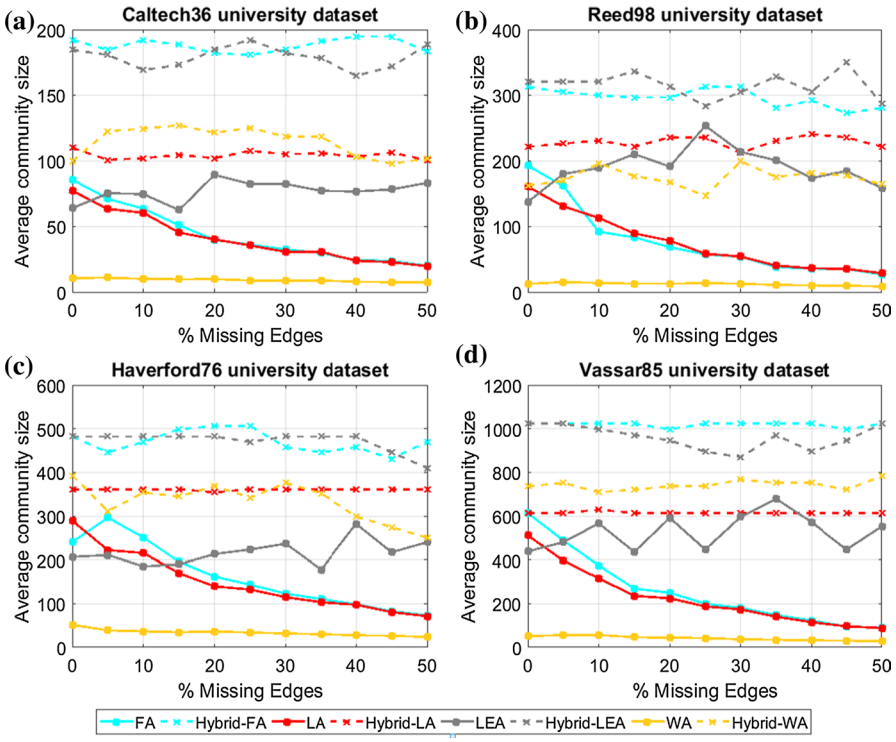


Fig. 8 Average community size for: **a** Caltech36 university dataset, **b** Reed98 university dataset, **c** Haverford76 university dataset, **d** Vassar85 dataset

6 Conclusion and future work

In this paper, an optimisation tool for the existing community detection algorithms is proposed. This tool could be used as a pre-processing stage that makes use of attribute information, shared neighbours and connectivity information aspects of the network to build a hybrid similarity matrix. Because the attributes in a network usually do not play equally important roles in clustering tasks, the proposed approach assigns a weighting value to each attribute during the process of building hybrid similarity matrix to reflect the relative importance of each attribute.

Besides the attribute weighting parameter, the approach required the specification of two more parameters α and β; these control the degree of contribution of connectivity information, attribute similarity and shared neighbours information for a good balance between them. The sensitivity of the pre-processing approach to α and β parameters is analysed. In addition, a simple but effective model for determining attribute weighting value and α and β values of the approach to achieve an optimal result is provided.

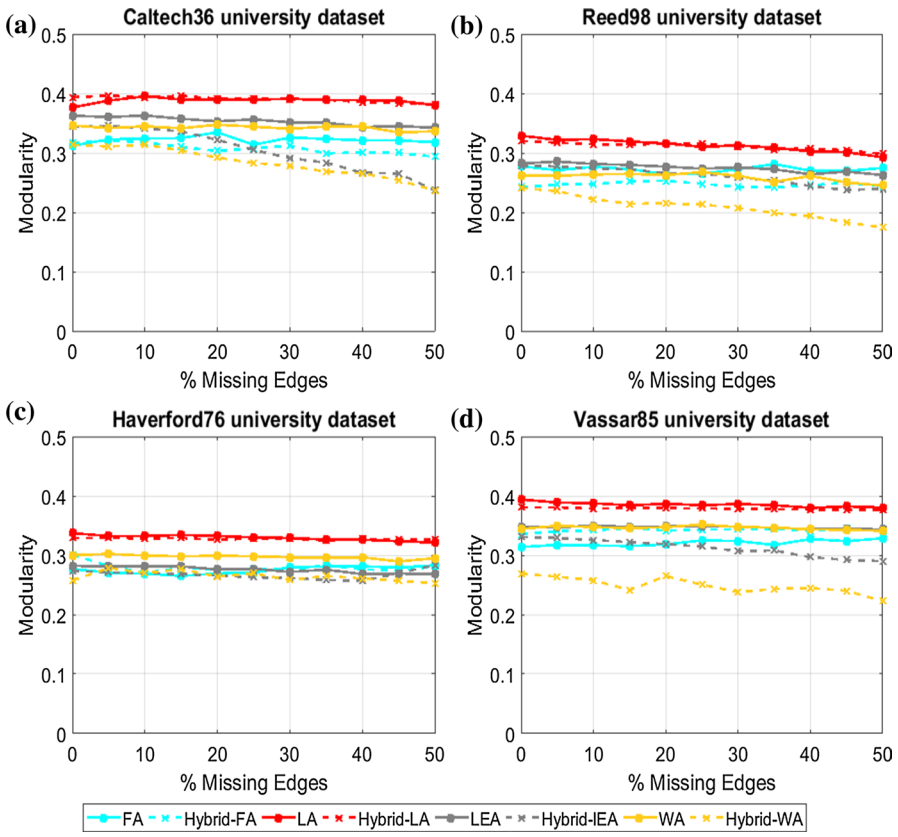


Fig. 9 Modularity index vs missing edges for: **a** Caltech36 university dataset, **b** Reed98 university dataset, **c** Haverford76 university dataset, **d** Vassar85 dataset

Table 2 Performance comparison of the proposed approach using different types of information

Algorithm	Caltech36 dataset		Reed98 dataset		Haverford76		Vassar85	
	%Missing edges		%Missing edges		%Missing edges		%Missing edges	
	0%	50%	0%	50%	0%	50%	0%	50%
Only link information	Good	Good	Good	Bad	Good	Bad	Good	Good
Only attribute information	Very Good	Bad	Bad	Bad	Good	Good	Good	Bad
Pre-processing link and attribute information	Good	Very good	Very good	Very good	Very good	Very good	Very good	Very good

A Jaccard similarity coefficient is used to denote attribute similarity between nodes and combined with adjacency matrix (links information). The approach is tested in conjunction with three traditional algorithms (Newman greedy algorithm, Louvain greedy algorithm and Neman spectral optimisation) popular in the literature by applying to three real-life Facebook data networks. Experimental results demonstrate that this approach yields better effectiveness and robustness than the state-of-the-art algorithms over noisy networks.

The proposed approach utilises a similarity function for comparing attributes. In a wide range of real-life applications, data contain a mixed type of attributes (e.g. numerical, categorical). Therefore, it is important to use appropriate similarity metrics to correctly measure the attribute proximity between two nodes in the network. However, the appropriate choice of the similarity measure depends on the attribute type of network to study. An interesting guideline to extend this research work is to use a more sophisticated approach that supports datasets with mixed attribute types. Furthermore, we have already developed a set of 'decentralised algorithms' for community clustering. We will be evaluating these algorithms with the pre-processing scheme proposed in this paper. We will also explore using the smartphone datasets (3.3 TB) collected by the University of Cambridge using Device Analyzer.

Acknowledgements The authors would like to thank their families for their continued support and encouragement during this work. In addition, they would like to thank the Libyan Cultural Bureau in London for their financial assistance in supporting the PhD work of the first author.

References

1. Mahata D, Patra C (2016) Detecting and analyzing invariant groups, in complex networks in computational intelligence in data mining—Volume 1. Springer, Berlin, pp 85–93
2. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3):75–174
3. Bhih A, Johnson P, Randles M (2019) Decentralized iterative approaches for community clustering in the networks. J Supercomput 75(8):4894–4917
4. Danon L et al (2005) Comparing community structure identification. J Stat Mech Theory Exp 2005(09):P09008
5. Orman GK, Labatut V, Cherifi H (2011) On accuracy of community structure discovery algorithms. arXiv preprint arXiv:1112.4134
6. Borgatti SP, Everett MG, Johnson JC (2013) Analyzing social networks. SAGE Publications Limited, Thousand Oaks
7. Schaeffer SE (2007) Graph clustering. Comput Sci Rev 1(1):27–64
8. Khatoon M, Banu WA (2015) A survey on community detection methods in social networks. Int J Educ Manag Eng (IJEME) 5(1):8
9. Warnke SD (2016) Partial information community detection in a multilayer network. Naval Postgraduate School Monterey United States, Monterey
10. Costa LDF et al (2007) Characterization of complex networks: a survey of measurements. Adv Phys 56(1):167–242

11. Gehweiler J, Meyerhenke H (2010) A distributed diffusive heuristic for clustering a virtual P2P supercomputer. In: 2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW). IEEE
12. Yu JY, Chong PHJ (2005) A survey of clustering schemes for mobile ad hoc networks. IEEE Commun Surv Tutor 7(1):32–48
13. Wang M et al (2015) Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework. Proc VLDB Endow 8(10):998–1009
14. Aggarwal CC, Wang H (2010) A survey of clustering algorithms for graph data. In: Managing and mining graph data. Springer, Boston, MA, pp 275–301
15. Newman M (2010) Networks: an introduction. Oxford University Press, Oxford
16. Newman ME (2004) Fast algorithm for detecting community structure in networks. Phys Rev E 69(6):066133
17. Blondel VD et al (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):P10008
18. Pons P, Latapy M (2006) Computing communities in large networks using random walks. J Graph Algorithms Appl 10(2):191–218
19. Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci 105(4):1118–1123
20. Newman ME (2001) Scientific collaboration networks. I. Network construction and fundamental results. Phys Rev E 64(1):016131
21. Davis JA (1970) Clustering and hierarchy in interpersonal relations: testing two graph theoretical models on 742 sociomatrices. Am Sociol Rev 35(5):843–851
22. Louch H (2000) Personal network integration: transitivity and homophily in strong-tie relations. Social Netw 22(1):45–64
23. Huijuan Z, Shixuan S (2013) A Graph Clustering algorithm based on shared neighbors and connectivity. In: 2013 8th International Conference on Computer Science & Education (ICCSE). IEEE
24. McPherson M, Smith-Lovin L, Cook JM (2001) Birds of a feather: homophily in social networks. Ann Rev Soc 27(1):415–444
25. Traud AL et al (2011) Comparing community structure to characteristics in online collegiate social networks. SIAM Rev 53(3):526–543
26. Bedi P, Sharma C (2016) Community detection in social networks. Wiley Interdiscipl Rev Data Min Knowl Discov 6(3):115–135
27. Plantié M, Crampes M (2013) Survey on social community detection. Social media retrieval. Springer, Berlin, pp 65–85
28. Zhou Y, Cheng H, Yu JX (2009) Graph clustering based on structural/attribute similarities. Proc VLDB Endow 2(1):718–729
29. Yang T, et al. (2009) Combining link and content for community detection: a discriminative approach. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM
30. Salem S, Ozcaglar C (2014) Hybrid coexpression link similarity graph clustering for mining biological modules from multiple gene expression datasets. BioData Min 7(1):16
31. Dang TA, Viennet E (2012) Community detection based on structural and attribute similarities. In: International conference on digital society (icds), pp 7–12
32. Ruan Y, Fuhry D, Parthasarathy S (2013) Efficient community detection in large networks using content and links. In: Proceedings of the 22nd International Conference on World Wide Web. ACM
33. Lin W, et al. (2012) Community detection in incomplete information networks. In: Proceedings of the 21st International Conference on World Wide Web. ACM
34. Wu P, Pan L (2016) Multi-objective community detection method by integrating users' behavior attributes. Neurocomputing 210:13–25
35. Traud AL, Mucha PJ, Porter MA (2012) Social structure of facebook networks. Phys A 391(16):4165–4180
36. Newman ME, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69(2):026113
37. Clauset A, Newman ME, Moore C (2004) Finding community structure in very large networks. Phys Rev E 70(6):066111
38. Newman ME (2006) Finding community structure in networks using the eigenvectors of matrices. Phys Rev E 74(3):036104

39. Pons P, Latapy M (2005) Computing communities in large networks using random walks. In: International Symposium on Computer and Information Sciences. Springer
40. Jaccard P (1902) Distribution comparée de la flore alpine dans quelques régions des Alpes occidentales et orientales. Bull de la Murithienne 31:81–92
41. Bhih A, et al. (2017) Decentralized iterative community clustering approach (DICCA). In: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). IEEE
42. Choi S-S, Cha S-H, Tappert CC (2010) A survey of binary similarity and distance measures. J Syst Cybern Informatics 8(1):43–48
43. Arif M, Basalamah S (2012) Similarity-dissimilarity plot for high dimensional data of different attribute types in biomedical datasets. Int J Innov Comput Inf Control 8(2):1275–1297
44. Leskovec J, Rajaraman A, Ullman JD (2014) Mining of massive datasets. Cambridge University Press, Cambridge
45. Bhih A (2018) High performance decentralised community detection algorithms for big data from smart communication applications. Liverpool John Moores University, Liverpool