

Article

Not peer-reviewed version

A Novel Method for Community Detection in Bipartite Networks

[Ali Movaghar](#)^{*}, [Ali Khosrozadeh](#), Mohammad Mehdi Gilanian Sadeghi, [Hamidreza Mahyar](#)

Posted Date: 24 March 2025

doi: 10.20944/preprints202503.1655.v1

Keywords: social networks; bipartite networks; community structure; community detection; voting



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

A Novel Method for Community Detection in Bipartite Networks

Ali Khosrozadeh ¹, Ali Movaghar ^{2,*}, Mohammad Mehdi Gilanian Sadeghi ¹
and Hamidreza Mahyar ³

¹ Department of Computer and Information Technology Engineering, Qa.C., Islamic Azad University, Qazvin, Iran

² Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

³ Faculty of Engineering, McMaster University, Hamilton, Ontario, Canada

* Correspondence: movaghar@sharif.edu

Abstract: The community structure is a major feature of bipartite networks, which serve as a typical model for empirical networks consisting of two kinds of nodes. Over the past years, community detection has drawn a lot of attention. Numerous methods for community detection have been put forth. Nevertheless, some of them need a lot of time, which restricts their use in large networks. While several low-time complexity algorithms exist, their practical value in real-world applications is limited since they are typically non-deterministic. Typically, in bipartite networks, a unipartite projection of one part of the network is created, and then communities are detected inside that projection using methods for unipartite networks. Unipartite projections may yield incorrect or erroneous findings as they inevitably include a loss of information. In this paper, BiVoting, a two-mode and deterministic community detection method in bipartite networks is proposed. This method is a consequence of bipartite modularity, which quantifies the strength of partitions and is based on how people vote in social elections. The proposed method's performance has been evaluated, and comparison with four common community detection methods in bipartite networks shows that for calculating the modularity score in large networks, BiVoting performs better than the best method.

Keywords: social networks; bipartite networks; community structure; community detection; voting

1. Introduction

Networks display various complicated systems in different disciplines. The nodes and edges of a network inferred from complicated systems represent entities and relationships. An edge connects a pair of nodes if they have a particular relationship. Numerous studies have focused on unipartite or one-mode networks that only feature one kind of node. In contrast, real networks usually have a variety of node kinds. A bipartite or two-mode network has two different kinds of nodes, and connections are limited to distinct node kinds. A bipartite network is a natural model for various real-world complex systems. A better method to comprehend modular structures in bipartite networks is through the concept of community. Community detection in complex networks can aid in the understanding of these networks structural characteristics. A straightforward approach to community detection was to apply the standard methods for community detection in unipartite networks after converting the bipartite networks into their unipartite counterparts. The issue of information loss in this procedure is a challenge. After The concept of bipartite modularity, in which two independent parts of nodes are placed independently in modular structures, was presented; the idea of community detection was proposed without the need to change bipartite networks into unipartite networks. The task of precisely maximizing modularity is typically unsolvable. Some useful approaches for determining the nodes' high modularity divisions have been presented. Centrality measures are commonly employed in social network analysis to ascertain the respective significance of nodes. HellRank is a precise metric for locating core nodes in bipartite networks. The fact that there is no need to transform the bipartite network to its unipartite counterpart makes this measure ideal.

Certain methods that have been suggested for community detection include issues such as excessive time and memory consumption, lack of assurance, inability to be applied in large-scale networks, and resolution issue limitations. Without having to be converted into a unipartite counterpart, the proposed method, which makes use of the ideas of bipartite modularity and the HellRank centrality measure, has been able to be implemented directly in large-scale bipartite networks. Also, the proposed method can significantly improve the calculation of modularity scores in networks, especially in large-scale networks. This is demonstrated by implementing the proposed method and four popular techniques for community detection on some artificial and real bipartite networks. After the first section, in Section 2, various methods for community detection in bipartite networks will be reviewed, and following Section 2, some of the preliminary definitions and variables that were used in this paper will be demonstrated. In Section 3, the proposed method, the relevant voting rules, and its algorithm are explained in detail. In Section 4, the results obtained from implementing the proposed method on several artificial and real networks are presented and compared with four well-known methods for detecting communities in bipartite networks.

2. Related Work

A variety of methods, including non-negative matrix decomposition [1], label propagation [2], expansion from seed sets [3], evolutionary method [4], game theoretical approaches [5], line graphs [6], and modularity optimization [7], have been suggested for unipartite community detection under various assumptions. One straightforward concept comes from the Zhou projection approach [8], which naturally adapts current community detection methods by converting bipartite networks into unipartite networks. However, scholars contend that projection approaches may result in incomplete information as, following projection, only a certain kind of node is kept while the other kind is lost [9]. Many methods have been devised to sustain two kinds of nodes following the division of communities. After expanding from unipartite modularity, Barber introduced bipartite modularity [10] and to impel two separate parts of nodes inside modular structures, BRIM was then created. It is noteworthy that bipartite network modularity has resolution limitations [11], as high modularity scores cannot reliably detect tiny communities. Lehmann introduced a method for bi-clique community detection, which is an extension of the k clique community detection algorithm [12]. Eigen-vectors of matrices [13], modularity optimization [14], clustering coefficient [15], intimate degree [16], stochastic block model [17], density-based modularity [18], and asymmetric intimacy [9] are just a few of the numerous methods with various assumptions in bipartite networks. Li [19] presented a unified community detection method based on the probability of node similarity to handle unipartite and bipartite networks concurrently. With the exception of unipartite networks, there should be a greater focus on community detection in big bipartite networks with a lot of nodes and edges using existing methods. Currently available methods often need a temporal complexity of at least quadratic, meaning that handling huge bipartite networks always takes several hours. The BiAttractor algorithm is an approach based on distance dynamics that is proposed for community detection in big bipartite networks [20]. It draws inspiration from the ways in which human civilization is structured, with more interactions taking place inside communities and less between them. The community detection problem is traditionally solved using graph partitioning and hierarchical clustering methods, like agglomerative algorithms and their divisive equivalents. The Newman concept of modularity has quickly acquired traction as a standard for evaluating partition quality [21]. The ratio of edges inside communities to edges between communities is known as modularity. Maximizing modularity using optimization approaches has been a prominent issue, as a high modularity value suggests a good partition [10]. There are several approximation methods to determine the precise best solution to the issue. Among them, Blondel [7] invented the Louvain algorithm, which is a greedy method. In [22], a two-stage method called the BiLouvain algorithm is presented for detecting community structure in bipartite networks.

Following, more than forty new studies in the field of community detection will be reviewed. In [23], a spectral clustering-based community detection method is proposed for community detection in multilayer networks. In [24], tensor low-rank representation, spectral clustering, and distance regularization are combined in a method to enhance the accuracy of community detection in multiplex networks. In [25], a parallel algorithm is proposed to solve the community detection overlapping problem in graph streams. In [26], a new algorithm for community detection based on different seed nodes (RWBS) is proposed, whose goal is to comprehend community structure as a novel idea for allocating network resources. RWBS employs modularity as an optimization function to join communities. In [27], a neural network-based method has been used to overcome the drawback that user data lacks temporal variation. Additionally, innovative techniques have been put forth to analyze community networks using a network embedding method based on the matrix factorization framework. In [28], a spectral clustering community detection algorithm, PMIK-SC, based on the point-wise mutual information graph kernel is proposed. Also, a quick approach called BI-CNE is suggested for determining the number of communities during spectral clustering. Link prediction, structure observation, network estimation, and community partitioning are the four parts of the LPGSE method, which was developed and put into practice in [29] to overcome the difficulty of extracting useful community information from the limited network structure. In [30], the goal is to recognize the community that contains a particularly important node called the anchor and to monitor how it changes over time. To effectively mine evolutionary community-aware information and to construct dynamic similarity metrics for dynamic link prediction, link prediction methods must take into account actor-level temporal changes and associated evolutionary information. Developing and integrating such dynamic features with machine learning algorithms is the goal of [31]. A community detection method in complex networks was proposed in [32] by combining Max-Min modularity and a connectivity-based metric at the same time. This allowed for the development of a novel complementary graph by utilizing the connectivity-based metric and a heuristic algorithm.

In [33], an improved method for community detection based on a scalable community fitness function is proposed. This method introduces a stringent strategy to filter the outputs and a new parameter to improve its scalability. In [34], an unsupervised community detection algorithm called LNSSL (Local Node Similarity-Integrated Stochastic Competitive Learning) is presented to overcome the community detection problem and enhance the stability and accuracy of stochastic competitive learning. A new online framework for concurrently learning node representations and detecting communities in a network termed Community Contrastive Learning (Community-CL) is proposed in [35]. In [36], a new algorithm that uses non-negative matrix factorization (NMF) to detect sparse network communities is presented. In [37], a new method based on parallel fuzzy cognitive maps (FCMs) is suggested to locate critical nodes. In order to detect communities that contain one or more illegal Bitcoin addresses, the research [38] first looks into the structure of Bitcoin transaction networks. After that, it gathers a list of illegal addresses and uses them to identify the communities that have been detected. In [39], a novel method for identify societal issues from web-based data using network analysis is proposed, wherein social issue candidates within the designated time frame are extracted using a community detection method. In [40], a new method (CSIM) is put forth by providing details about the community structure from the standpoint of edge information encoding. In [41], a new approach is presented that synergizes community detection algorithms with different graph neural network (GNN) models to reinforce link prediction in scientific literature networks. In [42], an improved version of the Link Entropy method is proposed to solve the challenge of measuring the significance of edges in a complex network. In [43], entropy-based measures were introduced as a new way to describe student's learning behaviors. By combining these measures with a proven community recognition technique, learning behaviors in student communities are analyzed.

In [44], a stochastic agglomerative algorithm to detect the local community of given seed nodes is proposed that has several superiorities, including hierarchical community structure detection, etc. In [45], the problem of estimating community memberships of nodes in directed networks is regarded.

Here, a directed degree-corrected mixed membership (DiDCMM) model is proposed. In [46], a method to calculate the number of communities for weighted networks produced from arbitrary distribution under the degree-corrected distribution-free model is proposed. In [47], unlike most studies, which consider the network topology to be a factor in community formation, the content of the network is considered. Here, a comprehensive online community detection method that simultaneously examines the network's structure and content is offered. In [48], a user similarity measure based on the network structure is proposed to improve recommendations to users with special preferences and new users, so that dealing with the gray sleep and cold start problems. In [49], a new method is suggested to construct and comprehend the structure of Reddit, which is based on crossposts that were cross-posted to another subreddit after first appearing on one. In [50], using NLP techniques such as text embeddings, subreddit content is directly modeled, and a subreddit graph network is built. In [51], an asymmetric ladder-shaped architecture called RGCLN based on multi-relational graph convolution that can create node representations with significant representational strength by fusing deep node properties is proposed. In [52], a method called local greedy extended dynamic overlapping community detection (GLOD) is proposed to deal with the difficulties of detecting overlapping communities in complex networks. In [53], for community detection, a new QUBO-based approach that only requires number-of-nodes qubits and is shown using a QUBO matrix is introduced. In [54], a collaborative approach is proposed based on matrix factorization and spectral embedding to recuperate the groups for the layers and nodes together. In [55], the local expansion technique based on ordinary graphs is extended to hypergraphs, and an efficient hypernetwork community detection method based on local expansion (LE) and global fusion (GF), called HLEGF, is proposed.

In [56], a new Community-Enhanced Contrastive Learning method is introduced to help the recommendation primary task called CECL, which uses a community detection algorithm in the bipartite graph. In the traditional LPA, every node is considered to have an equivalent relationship, and unreliable nodes reduce the label propagation accuracy. Therefore, in [57], the influence-based community overlap propagation algorithm (INF-COPRA) to rank the labels and nodes influence is proposed. The community detection method has been emphasized as a practical technique for handling the LSGDM complexity. However, dealing with the reliability, hesitancy of information, and interpretability of the method is still challenging. Therefore, in [58], a new approach of a Z-hesitant fuzzy network with the implementation of the community detection method is introduced. Orthogonal non-negative matrix tri-factorization (ONMTF) displays noteworthy potential as a strategy for this. In [59], the application of ONMTF in multiplex networks is investigated with the goal of concurrently detecting exclusive and shared communities. In [60], a modularity function F_2 as a new objective function is proposed. Modularity function F_2 overcomes certain drawbacks of the previous modularity functions, such as the resolution limit. In [61], a community detection method based on node influence analysis is proposed, wherein Pareto dominance is employed to rank the node's influence. In [62], a multi-objective community detection technique based on a pigeon-inspired optimization algorithm, MOPIO-Net, is proposed. In [63], a new centrality measure called density-based entropy for identifying the most significant nodes locally is proposed, which offers an efficient way to locally identify the most significant nodes and community detection. Community detection models based on non-negative matrix factorization are superficial and unable to completely recognize the internal organization of complex networks. So in [64], a new limited symmetric non-negative matrix factorization with deep autoencoders (CSDNMF) as a solution is introduced. In [65], a model based on convex game theory and a measure of community strength is proposed to tackle the problem of identifying meaningful communities.

2.1. Preliminary Definitions and Variables

2.1.1. Community Structure

A sub-graph whose nodes are more likely to link to one another than to nodes outside the sub-graph is called a network community. One of the key features of bipartite networks is their community structures, which divide networks into collections of nodes and edges. With tight connections between

nodes in the same group and weak connections between groups of distinct nodes, the underlying networks are organized in a modular fashion. Coarse-grained sub-networks may be detected from underlying networks. Global average attributes are inadequate to describe the community structure features since various communities have varied structural characteristics [14]. Moreover, the community offers an improved comprehension of modularity in bipartite networks. There are two categories into which bipartite communities can be divided. Members of a bipartite network are categorized as belonging to the same type or to separate types. In bipartite networks, the community structure combines comparable nodes within the same part and shows the extent to which nodes in one part are connected to other part's nodes.

2.1.2. Centrality Measure

Centrality measures are frequently used in social network analysis to express the respective significance of nodes. It is possible to identify prominent users using social network analysis by applying this idea of centrality [66]. Numerous centrality measures, such as Betweenness, Degree, and Closeness, have been put forth to identify network nodes. Utilizing these structural network measures necessitates a thorough comprehension of the network topology [67]. In order to transform bipartite networks into unipartite networks, some approaches have been applied to bipartite networks using various projection techniques. A few of the unipartite network measures have been applied to bipartite networks, despite the fact that unipartite networks are less informative than their bipartite counterparts [68]. Conventional centrality measures are problematic in large-scale social networks because they need a thorough grasp of network topology and because there are no suitable measures for detecting more behaviorally representative individuals inside bipartite social networks. A novel centrality measure known as HellRank was created by taking into account users significance indications while detecting core nodes [69]. Without initially generating a unipartite projection of the bipartite network, HellRank is a measure that may be applied to community detection in bipartite networks. One of the reasons for using this measure in the proposed method is this. The Hellinger distance, a kind of f-divergence metric, is the basis of HellRank and shows how similar a node is structurally to the other nodes. Empirical data indicates that nodes in bipartite networks exhibiting high HellRank also have high Degree, Betweenness, and PageRank. A function that quantifies the difference between probability distributions P and Q is called a f-divergence, and its value is $D_H(P||Q)$. For a convex function f and $f(1) = 0$, the f-divergence of Q from P is defined as follows:

$$D_f(P||Q) = \int_{\Omega} f\left(\frac{dP}{dQ}\right) dQ \quad (1)$$

Where the set of all potential results is denoted by Ω . The Hellinger distance for discrete probability distributions, $P = (p_1, p_2, \dots, p_m)$ and $Q = (q_1, q_2, \dots, q_m)$, where m is the vector's length, is as follows:

$$D_H(P||Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^m (\sqrt{p_i} - \sqrt{q_i})^2} \quad (2)$$

This is clearly connected to the Euclidean norm of the vector difference square root, as:

$$D_H(P||Q) = \frac{1}{\sqrt{2}} \|\sqrt{P} - \sqrt{Q}\|_2 \quad (3)$$

2.1.3. Modularity

The modularity measure, first proposed by Newman [21], quantifies the degree to which edges emerge inside communities rather than across communities. We can evaluate the quality of any node-to-community assignment by utilizing modularity. The task of precisely maximizing modularity is typically unsolvable. A number of useful approaches for determining the nodes' high modularity divisions have been presented. It is also possible to define modularity by employing a matrix whose

eigen-spectrum is directly associated with the modularity of the network. Given that modularity is explicitly dependent on a null model, the choice of null model has a significant influence on modularity. In statistical physics, communities in bipartite networks are found by first building a unipartite projection of one part of the network, then using community detection techniques related to working with unipartite networks. Unipartite projections are inherently less informative, notwithstanding their potential for usefulness. Barber offered a concept of modularity for bipartite networks based on creating a bipartite modularity matrix, which was an extension of the Newman work. These matrix eigen-spectrum features have been identified, and the Newman method has been tailored for bipartite networks using these features. For bipartite, recursively induced modules, this technique is referred to as BRIM [70].

3. Proposed Method

Getting precise and high-quality community structures out of networks is still a difficult undertaking. The proposed method is significant because it can definitely detect distinct and high-quality community structures from networks. First, the voting patterns seen in election activities used to model the network's voting process. Every node votes for the candidates that have been nominated using the suggested voting criteria, and densely linked groups of nodes can rapidly come to an agreement on their choices. This process creates a cluster of candidates and their supporters. In order to create the final community structure, the clusters will serve as the first communities, and some of them will be effectively combined into larger ones. It was discovered through a thorough examination of community structures taken from several networks that the node and the majority of its neighbors are members of the same community. Furthermore, every node in a community forms a tiny cluster with its neighbors. Every community consists of several tiny clusters. There are always some nodes with a comparatively greater degree than others in every cluster. A cluster is a collection of nodes connected to a single, higher-degree node. In social systems, voting behavior is comparable to this phenomenon. There are always powerful local figures in networks that are detached from social systems, such as department heads and experts in certain domains. Assuming those people take part in an election where nominations are open to everybody, each voter will break ties by selecting the person nearest to them and casting their vote for the person who has more influence around them. A node group will be formed as a result of this process. The proposed method for community structure detection in networks is based on simulating the voting process. The approach that is being given uses a few predefined principles as voting guidelines, uses degree to indicate each node's impact, and uses similarity between nodes to indicate how close they are. Stated differently, every node casts a vote for a neighbor whose degree exceeds its own. The voter node chooses the larger-degree node with the greatest similarity if there are many of them with the same impact among its neighbors. As a result, every node votes rapidly, and at the conclusion of the voting process, we will have a large number of little node clusters. Every community consists of many clusters. The clusters serve as the starting communities, which combine to form the final community structure.

3.1. Voting Rules

This study deals with a graph without direction or weight $G = (U, I, E)$, where two sets of distinct node kinds are represented by U and I , and various node kinds are connected by the edge e of E . There are no edges connecting any of the nodes in the same set. The proposed method simulates election voting patterns to detect groups inside networks. An important part of the detection procedure is the voting rules. Each node $u \in U$ votes throughout the detection procedure in accordance with the following guidelines:

1. If node u has the highest degree among its second-order neighbors or has been nominated as a candidate, it casts a vote for itself.
2. From among u 's second-order neighbors, a node with a greater degree than u is selected and designated as v . If there are many nodes that are similar to u , the most similar node is chosen and

designated as v . The notation $sim(u, v)$ indicates the similarity between u and v . u proposes itself as a candidate and casts its own vote if $sim(u, v) = 0$.

3. If v has not cast a vote for any other nodes, u proposes node v as a candidate and casts a vote for it.

4. If node v casts a vote for node w , i.e., if v has forfeited its right to be nominated as a candidate, then node u will also cast a vote for node w .

The voting arrangement of nodes can affect the arrangement in which nodes are nominated as candidates, as per the voting rules mentioned above. As a result, various voting orders may produce different outcomes. The clustering coefficient of each node in the set U is computed according to equation 4, and the nodes are then voted in ascendant order based on their clustering coefficients. For any node $x \in U$, clustering coefficient can be computed as a portion of the number of 4-paths centered on the focal node x to the subset of these in which the path's first and last nodes share a common node that is not part of the 4-path (part of at least one 6-cycle) and can be calculated as follows [68]:

$$cc(x) = \frac{\text{closed 4-paths centered on node } x}{\text{4-paths centered on node } x} \quad (4)$$

The higher the number of edges that a certain node v has with its neighbors, the higher the clustering coefficient x . $cc(x)$ achieves its maximum, 1, whenever the sub-graph made up of node x 's second-order neighbors is a complete graph. However, in a sparse network, it is improbable that any node's neighborhood sub-graph will be a full graph, particularly for nodes with higher degrees. Unless stated otherwise, a node with a higher degree always has a small clustering coefficient. The nodes with greater degrees can nominate as a candidate early and surround themselves with their neighbors to create clusters by voting based on the ascendant order of the node's clustering coefficients. Voting rule 2 states that the similarity between a node and u is utilized to decide which node u should vote for if there are many nodes in the neighborhood with degrees higher than u . As a result, voting heavily relies on node similarity. In the proposed method, the similarity between nodes u and v determined using the following formula:

$$sim(u, v) = \frac{1}{\text{hellinger distance}(u, v)} \quad (5)$$

Following the voting process, we will have several little clusters. These clusters deviate from the intended communities nonetheless. Several of these clusters may be present in a final community, based on the data mentioned above. Consequently, we regard these clusters as the starting communities and contemplate combining some of them to create the final communities. Only some communities take into considered as a pairings of communities that have a similarity higher than a certain level in order to guarantee high-quality results. The similarity between two communities C_i and C_j , computed as follows:

$$sim(C_i, C_j) = \sum_{x \in U_{C_i}} \sum_{y \in U_{C_j}} \frac{1}{\text{hellinger distance}(x, y)} \quad (6)$$

There are two stages to the proposed method, which is known as the BiVoting. The voting process, which produces a number of little clusters, is the first stage. In order to build the final community structure, the smaller clusters will merge into bigger ones during the merging phase as a second stage. Two stages of the proposed method are displayed in Figure 1, and Algorithm 1 provides a description of the method framework.

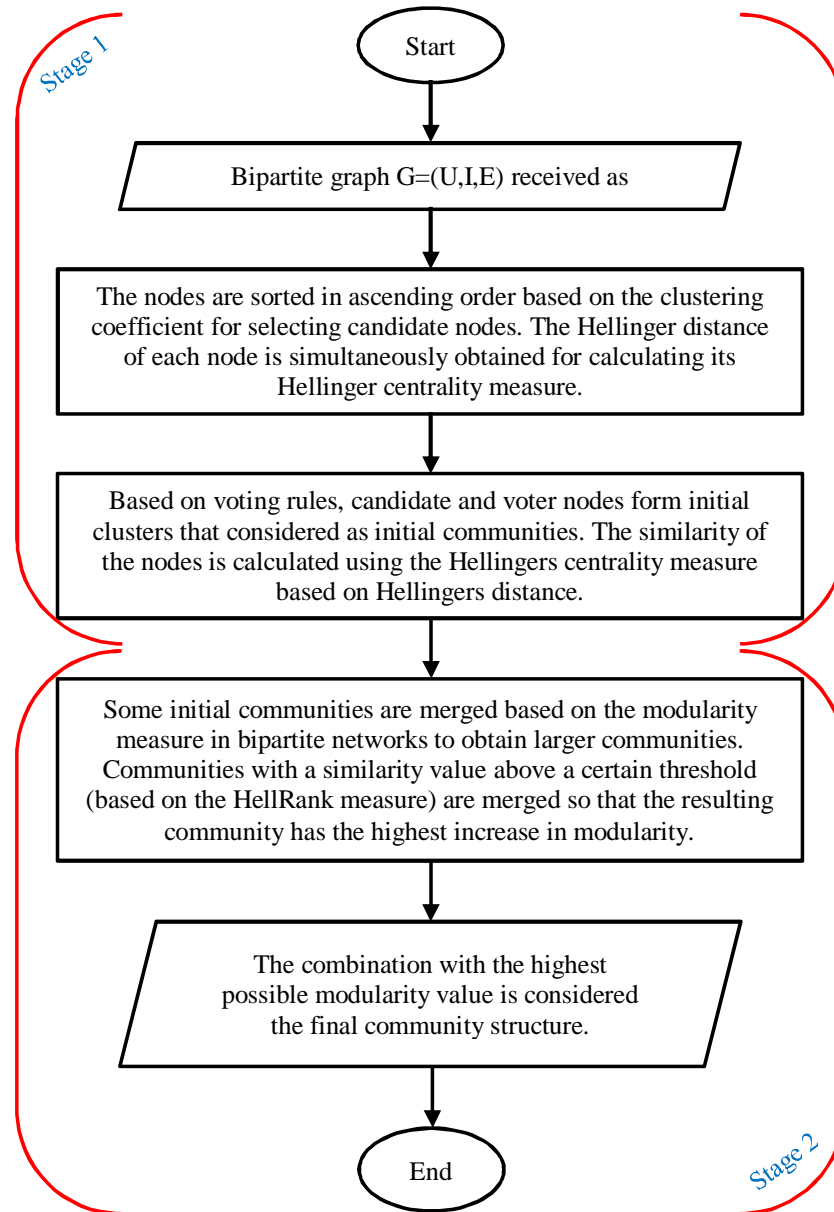


Figure 1. Two stages of the proposed method.

Algorithm 1: BiVoting Community Detection Method**Data:** $G = (U_{sers}, I_{tems}, E)$, bipartite network**Result:** CS, the final community structure

```

1 begin
2    $CS_{init} \leftarrow \text{BiVoting}(G)$ 
3    $CS \leftarrow \text{BiMerge}(G, CS_{init})$ 
4   return CS
  
```

The BiVoting() function is equivalent to the voting process, wherein, according to the voting rules, every node in the network casts a vote for the node that has a bigger impact in its neighborhood. The reasoning of Algorithm 2 is given in pseudo-code, and several little clusters are the result. It should be mentioned that many of these little clusters can be combined to form a single community; they are not the final communities. The tiny clusters are used as the starting communities in the resultant community structure, and some communities are merged via the function BiMerge(). In the proposed method, using an approach similar to that of [70], this function is implemented by merging

two communities whose merger would provide the biggest modularity increase in each iteration. The merging procedure pseudo-code is explained in Algorithm 3. This merging procedure results in a substantially lower joining time since each tiny cluster is regarded as an initial community instead of each node being a community. Also, the merging of two communities is only considered if their similarity exceeds a certain level, because doing so may have an adverse effect on the quality of the new community structure. Equation 6 is used for determining the similarity between two communities. Instead of continuing until every node is in the same community, the joining process can potentially end sooner if there isn't similarity between any two communities that is higher than a predetermined threshold. As a result, the merging process is quite effective.

Algorithm 2: The Logic of Function BiVoting(G)

Data: $G = (U, I, E)$, bipartite network
Result: CS_{init} , initial community structure

```

1 begin
2   for  $\forall x, y \in U$  do
3      $cc(x) \leftarrow \frac{\text{closed 4-paths centered on node } x}{\text{4-paths centered on node } x}$ 
4     calculate hellinger distance( $x, y$ ) from hellinger matrix of users
5    $V_u \leftarrow U, i \leftarrow 1, CS_{init} \leftarrow \emptyset$ 
6    $cand \leftarrow [0, 0, \dots, 0]_u, vote \leftarrow [0, 0, \dots, 0]_u$ 
7   while  $V_u \neq \emptyset$  do
8      $u \leftarrow \text{argmin}_x \{cc(x) | x \in V_u\}$ 
9      $V_u \leftarrow V_u - \{u\}$ 
10    if  $cand[u] \neq 0$  then
11      continue
12     $LDN(u) \leftarrow \{v | |N(v)| > |N(u)|, v \in NN(u)\}$ 
13    if  $LDN(u) = \emptyset$  then
14       $cand[u] \leftarrow i, vote[u] \leftarrow i$ 
15       $C_i \leftarrow \{u\}, CS_{init} \leftarrow CS_{init} \cup \{C_i\}$ 
16       $i \leftarrow i + 1$ 
17      continue
18    for  $\forall z \in LDN(u)$  do
19       $sim(u, z) \leftarrow \frac{1}{\text{hellinger distance}(u, z)}$ 
20     $v \leftarrow \text{argmax}_z \{sim(u, z) | z \in LDN(u)\}$ 
21    if  $sim(u, v) = 0$  then
22       $cand[u] \leftarrow i, vote[u] \leftarrow i$ 
23       $C_i \leftarrow \{u\}, CS_{init} \leftarrow CS_{init} \cup \{C_i\}$ 
24       $i \leftarrow i + 1$ 
25      continue
26    if  $vote[v] = 0$  then
27       $cand[v] \leftarrow i, vote[u] \leftarrow i, vote[v] \leftarrow i$ 
28       $C_i \leftarrow \{u, v\}, CS_{init} \leftarrow CS_{init} \cup \{C_i\}$ 
29       $i \leftarrow i + 1$ 
30    else
31       $j \leftarrow vote[v], vote[u] \leftarrow j$ 
32       $C_j \leftarrow C_j \cup \{u\}, \text{update } C_j \in CS_{init}$ 
33  return  $CS_{init}$  and information about items that linked with users in  $\forall C_i \in CS_{init}$ 

```

Algorithm 3: The Logic of Function BiMerge(G, CS_{init})**Data:** $G = (U, I, E)$, bipartite network; CS_{init} , initial community structure**Result:** CS , the final community structure

```

1 begin
2    $CS \leftarrow CS_{init}$ 
3    $Q \leftarrow \frac{1}{m} \sum_{i=1}^{|CS|} (\Psi_{C_i}^U - \frac{\Psi_{C_i}^U \cdot \Psi_{C_i}^I}{m})$ 
4    $Q_{max} \leftarrow Q$ 
5   while  $|CS| > 1$  do
6      $\Delta Q_{max} \leftarrow -2$ 
7     for  $\forall C_i, C_j \in CS$  do
8       if  $\sum_{x \in U_{C_i}} \sum_{y \in U_{C_j}} \frac{1}{\text{hellinger distance}(x,y)} \geq a \text{ threshold}$  then
9          $\Delta Q \leftarrow \sum_{\forall \text{ User} \in C_j \text{ and Item} \in C_i} (\frac{k_{C_i,p}^I}{m} - \frac{k_p \cdot \Psi_{C_i}^I}{m^2})$ 
10        if  $\Delta Q > \Delta Q_{max}$  then
11           $\Delta Q_{max} \leftarrow \Delta Q$ 
12           $(r, t) \leftarrow (i, j)$ 
13      if  $\Delta Q_{max} = -2$  then
14        break
15       $C_{rt} \leftarrow C_r \cup C_t$ 
16       $CS \leftarrow CS \cup C_{rt} - \{C_r, C_t\}$ 
17       $Q \leftarrow Q + \Delta Q_{max}$ 
18      if  $Q > Q_{max}$  then
19         $Q_{max} \leftarrow Q$ 
20         $CS_{max} \leftarrow CS$ 
21   $CS \leftarrow CS_{max}$ 
22  return  $CS$ 

```

4. Experimental Results

In this section, BiVoting will be compared with four popular methods for detecting communities in artificial and real bipartite networks. A quick explanation of these methods is provided below. BiVoting code was created in the Colab environment, and results were achieved by implementing this code on an Intel E5-2683 v4 Broadwell @2.1GHz server.

1. BiAttractor, Based on distance dynamics, Hong Sun created a unique technique to bipartite community detection in big bipartite networks [20]. The interactions that take place in human societies—where there are more interactions inside a society than between them—are the source of inspiration for it. It produces accurate community partitions and has $O(|E|)$ time complexity in networks that are sparse (the $|E|$ is edge number).

2. Adaptive BRIM, Barber introduced the concept of iteratively maximizing modularity Q_b in bipartite networks, which led to the creation of BRIM (bipartite, recursively driven modules) [70]. It is ensured that Q_b does not drop for each iteration. On the other hand, the observed splitting of bipartite networks results in a local maximum Q_b instead of a global maximum. Furthermore, the maximization of modularity Q_b also determines the number of modules. It has $O(cn)$ time complexity.

3. LP BRIM, By putting out the combination BRIM and label propagation technique known as LP BRIM, Liu has expanded on the work of BRIM [71]. It may be implemented in real networks because of its time complexity, which is $O(n^2)$ (where the number of nodes is n).

4. AsymIntimacy, Considering the intimate degree between nodes of the same kind and nodes of different kinds, Wang established asymmetric parameters [9]. Because of the close degree of asymmetry, nodes of the same kind are first combined as subsets. To form core communities, the subsets from the

previous stage are then combined with another sort of node. Core community pairings are combined as soon as the intersection ratio rises over a certain level. Until no other core communities can be combined, this procedure is repeated. $O(2n^2 + mn)$ is its time complexity, where n and m are the number of nodes and edges.

In general, two measures may be employed to examine and contrast various approaches. Normalized Mutual Information (NMI) is used to compute a score in the range $[0,1]$ when community splits across the underlying network are known beforehand. If not, modularity is used [10]. Modularity was first restricted to unipartite networks. Barber [70] has therefore expanded the concept of the community to include Q_b . There are more edges within the same community than predicted by the null model when the value from $[0,1]$ is greater. Yet, because of the resolution limit [11], Q_b has some restrictions.

4.1. Complexity Analysis

An algorithm's low temporal complexity is beneficial when it comes to applying it to large-scale networks. There are two stages to the proposed method. To acquire the first communities, the voting process is simulated in the first stage. $O(md + n \log n)$ may be used to achieve it, where d is the average degree of nodes and n and m are the number of nodes in one of two distinct sets of nodes and edges, respectively. The final communities are acquired in the second stage, which has a complexity of $O(mk)$, where k is the cost of merging iterations and $k \ll n, d < k$ in general. Some of the initial communities are combined to generate the final communities. For sparse networks, this means that the proposed method's overall time consumption is $O(md + n \log n + mk) \sim O(mk)$, or $O(nk)$. As a consequence, the experimental findings show that the proposed method is effective when used with large-scale networks. A comparison of the time complexity of the proposed method and the four compared methods in Table 1, shows that the proposed method is either better or at an equal level.

Table 1. Comparison of BiVoting and other methods on time complexity.

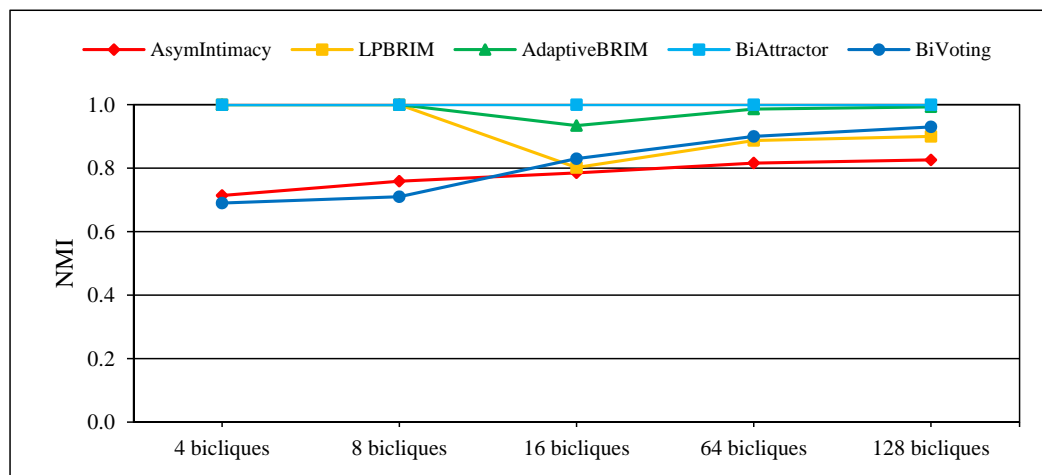
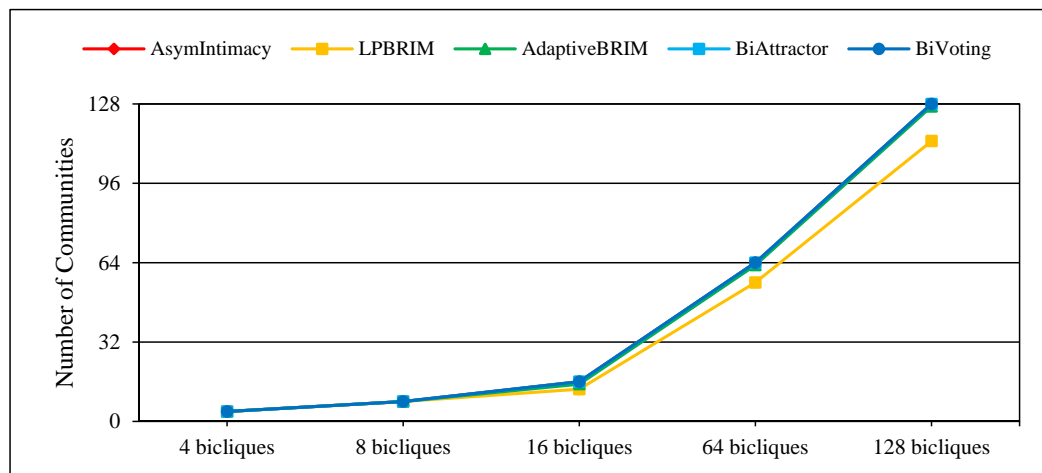
Method	AsymIntimacy	LP BRIM	Adaptive BRIM	BiAttractor	BiVoting
Time Complexity	$O(2n^2 + mn)$	$O(n^2)$	$O(cn)$	$O(E)$	$O(nk)$

4.2. Results in Artificial Networks

Numerous techniques for detecting communities rely on optimizing modularity. Nevertheless, the resolution limit issue may arise with these techniques [11]. Considering the network's total size and internal connections, they are limited in their ability to detect communities on less than a certain scale. Rings of bicliques with various biclique numbers have been created to show off the usefulness of BiVoting. Three squares are used to fully link two circles, and each biclique contains two mode nodes. Table 2 provides an overview of these rings of bicliques fundamental topological structural characteristics. Additional tests are conducted on rings with varying quantities of bicliques. In-depth findings are displayed in Figures 2 and 3, where NMI denotes the degree of accuracy of various bipartite community detection methods and N_C represents the total number of communities detected, respectively. The results show that the proposed method, like BiAttractor, can accurately detect the number of communities despite calculating a lower NMI value, which is due to the ability of BiVoting to accurately replicate the voting process. It should be noted that as the biclique size increases, the calculated NMI value gradually approaches the best value. These observations confirm that BiVoting outperforms other methods.

Table 2. Fundamental topological characteristics of rings of bicliques [20].

Bicliques	n	m	N	M	$\langle K \rangle$	C
4bicliq.	12	8	20	28	2.8	0.482
8bicliq.	24	16	40	56	2.8	0.482
16bicliq.	48	32	80	112	2.8	0.482
64bicliq.	192	128	320	448	2.8	0.482
128bicliq.	384	256	640	896	2.8	0.482

**Figure 2.** Performance of BiVoting and other methods on rings of bicliques for measurement NMI.**Figure 3.** Performance of BiVoting and other methods on rings of bicliques for measurement N_C .

4.3. Results in Real Networks

Real bipartite networks with unknown modular structures were used for experimental research, and the modularity score (Q_b) was used to confirm correctness. The following real bipartite networks are included in our study: S.W., A.R., S.C.I., C.N., M.G., P.C.D., D.W., and D.P. Following is a brief explanation of these real networks. Also, Table 3 provides a summary of their fundamental topological structural properties.

1. Southern Women (SW), In the context of bipartite network community detection, it is a widely recognized benchmark data set. Davis collected the information in and around Natchez, Mississippi in the 1930s in order to research race and class. It details how 18 women were divided up across 14 social gatherings. A bipartite network with 32 nodes and 89 edges is formed in this data set by women and the social activities they attend.

2. American Revolution (AR), 136 people's membership details from five organizations that date back to the American Revolution are included in the data collection [72]. It is possible to think of the connection between well-known people and their organizations as a bipartite network. An edge indicating membership in an organization can be found between an individual and that organization.

3. Scotland Corporate Interlock (SCI), Board members of Scottish corporations who served as directors on several occasions during 1904 and 1905 are included in this data collection. A bipartite network including 108 companies and 136 persons is upheld via directorships.

4. Crime Network (CN), People who have been a victim, witness, or suspect in at least one criminal case are included in this data set [72]. A bipartite network with 1476 edges linking 829 individuals and 551 criminal cases is formed by the interactions between crime-related individuals and crime cases.

5. Malaria and var Genes (MG), Using protein camouflages expressed in var genes, the malaria parasite eludes the human immune system. Regular recombination of Var genes results in a restricted genetic sub-string that generates novel camouflages [17]. As a result, var genes and their genetic sub-strings are linked in a way that creates a bipartite network with an inherent community structure. The 297 genes and 806 sub-strings that are connected by 2965 edges make up the MG network.

6. The Protein Complex and Drug (PCD), Meaningful links between acute illnesses and protein complexes have been found in recent investigations. In order to determine the inter-connectivity in systems relevant to human and molecular diseases, Nacher and Schwartz examined a bipartite network including 680 medications and 739 protein complexes [73]. PCD consists of 3690 edges and 1419 nodes.

7. DBpedia Writer (DW), DW has a lot of nodes and edges, making it difficult for current techniques to detect community patterns. The 46215 authors and 89356 works from DBpedia make up the DBpedia Writer network. The partnerships among writers to develop works are represented by the 144342 edges [72].

8. DBpedia Producer (DP), DBpedia is the source of the DBpedia Producer network. 48833 producers and their 138839 works make up this bipartite network [72].

Table 3. Fundamental topological characteristics of real bipartite networks [20].

Network	n	m	N	M	$\langle K \rangle$	C
S.W.	18	14	32	89	5.563	0.328
A.R.	136	5	141	160	2.270	0.781
S.C.I.	108	136	244	358	3.140	0.303
C.N.	829	551	1380	1476	2.139	0.427
M.G.	297	806	1103	2965	5.376	0.227
P.C.D.	680	739	1419	3690	1.746	0.407
D.W.	89356	46215	135571	144342	2.129	0.447
D.P.	48833	138839	187672	207268	2.209	0.514

Figures 4–6 display the comprehensive outcomes of the experiments conducted on the above small and medium real bipartite networks. Figures 7–9 also show the results of the experiments on the large real bipartite networks. In these figures, Q_b represents the modularity score of the various bipartite community detection methods, the number of communities detected is denoted by N_C , and the execution time of bipartite community detection methods is represented by $Time(s)$. In fact, the modularity score indicates the accuracy of a method for community detection, which can be used to evaluate the quality of each node-to-community assignment. Figures 4 and 7 contrast BiVoting's accuracy with other methods. Given that BiVoting and the other methods under comparison maximize modularity, we can deduce that BiVoting performs better than the other four methods when we see that its calculated modularity score is higher than that of BiAttractor and the other three methods under comparison. In BiVoting, although fewer communities are found, this reduction improves as the network gets bigger. Additionally, it must be noted that fewer communities are

detected due to BiVoting’s ability to accurately replicate the voting process. The comparison charts of community detection methods on large real networks are separated from smaller real networks because in extremely big networks like DW and DP, AsymIntimacy, LPBRIM, and AdaptiveBRIM methods cannot be used, and after many hours of execution, they either fail or receive no specific result. BiVoting executes faster on small real networks like SW, AR, SCI, and CN than other methods, but in medium real networks like MG and PCD, it doesn’t perform faster than all methods. Although in large real networks like DW and DP, compared to the BiAttractor method, the proposed method is implemented and executed in an acceptable time difference, which is a significant advantage given the large scale of the networks. These findings support the notion that Bivoting is a suitable technique for community detection in bipartite networks.

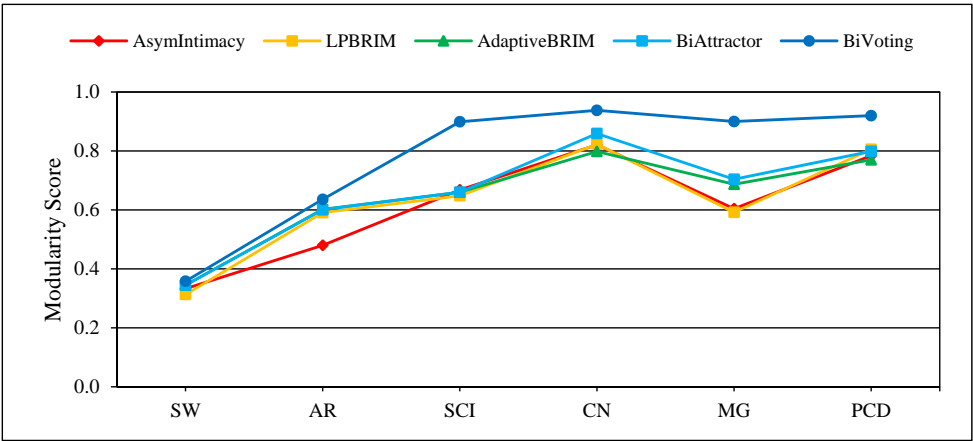


Figure 4. Performance of BiVoting and other methods on small and medium real networks for measurement Q_b .

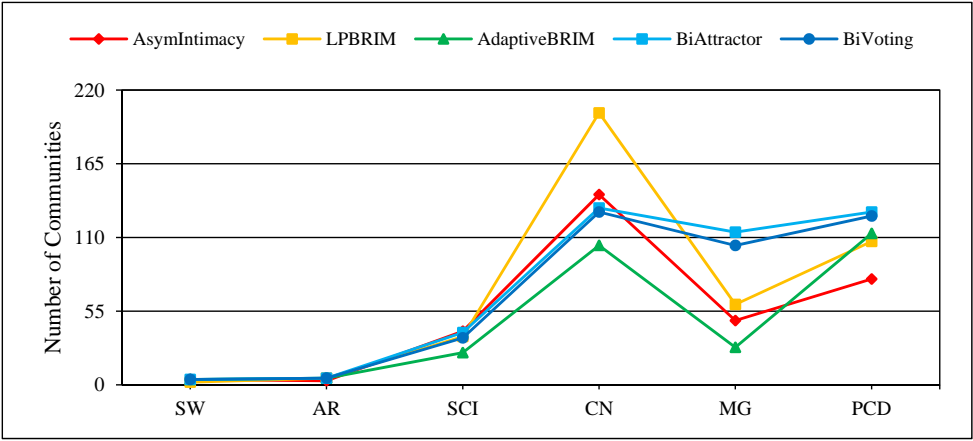


Figure 5. Performance of BiVoting and other methods on small and medium real networks for measurement N_C .

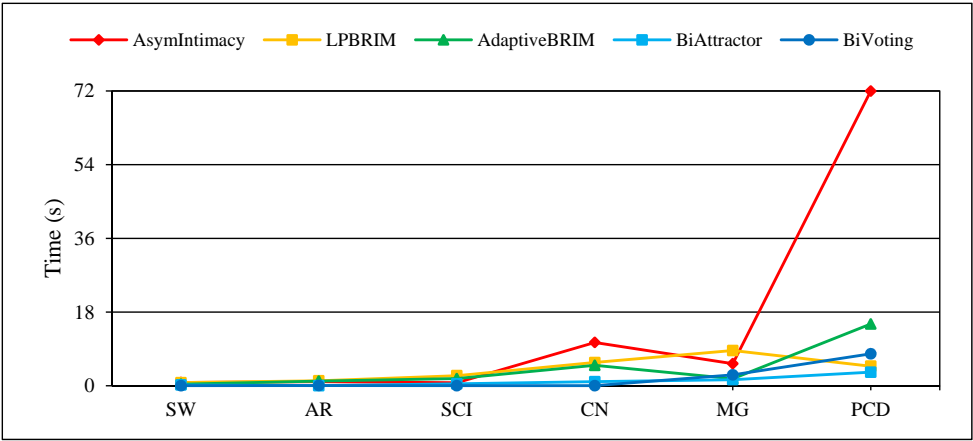


Figure 6. Time consumption comparison of BiVoting and other methods on small and medium real networks.

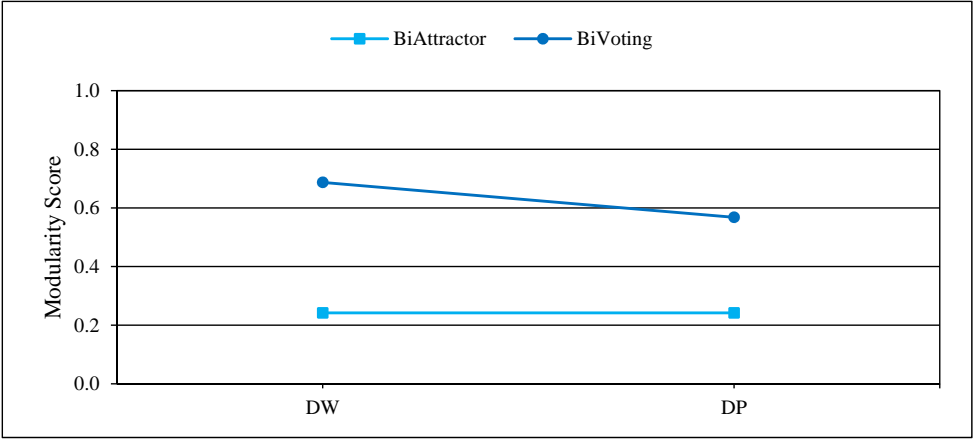


Figure 7. Performance of BiVoting and BiAttractor on large real networks for measurement Q_b .

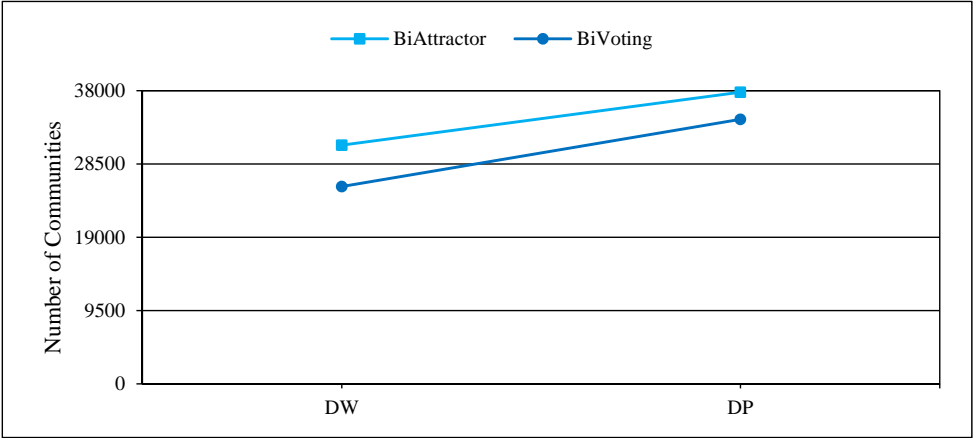


Figure 8. Performance of BiVoting and BiAttractor on large real networks for measurement N_C .

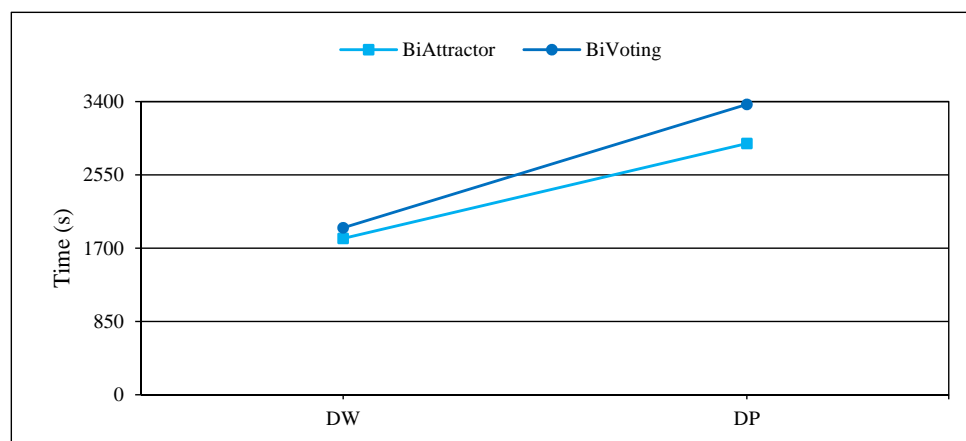


Figure 9. Time consumption comparison of BiVoting and BiAttractor on large real networks.

5. Conclusion and Future Work

A bipartite network is a natural model for various real-world complex systems. A better method to comprehend modular structures in bipartite networks is through the concept of community. Community detection in complex networks can aid in the understanding of these networks structural characteristics. Numerous methods for community detection have been put forth. Nevertheless, some of them need a lot of time, which restricts their use in large networks. While several low-time complexity algorithms exist, their practical value in real-world applications is limited since they are typically non-deterministic. A straightforward approach to community detection was to apply the standard methods for community detection in unipartite networks after converting the bipartite networks into their unipartite counterparts. The issue of information loss in this procedure is a challenge. After present the concept of bipartite modularity, in which two independent parts of nodes are placed independently in modular structures, the idea of community detection was proposed without the need to change bipartite networks into unipartite networks. Also, since the task of precisely maximizing modularity is typically unsolvable, some useful approaches are presented to determine the exact node's modularity divisions. Centrality measures are commonly employed in social network analysis to ascertain the respective significance of nodes. HellRank is a precise metric for locating core nodes in bipartite networks. The fact that there is no need to transform the bipartite network to its unipartite counterpart makes this measure ideal.

In this paper, BiVoting, a two-mode and deterministic community detection method in bipartite networks, was proposed, which was defined with the help of two concepts of bipartite modularity and HellRank centrality measure and based on the simulation of the election voting patterns in human societies. The proposed method's performance has been evaluated, and the results of comparison with four common community detection methods on variable-size artificial and real bipartite networks show that BiVoting performs better than other methods. The results of tests on rings with varying quantities of biclique show that the proposed method, like BiAttractor, can accurately detect the number of communities despite calculating a lower NMI value, which is due to the ability of BiVoting to accurately replicate the voting process. It should be noted that as the biclique size increases, the calculated NMI value gradually approaches the best value. The outcomes of the experiments conducted on the real bipartite networks show that BiVoting can calculate the higher modularity score than other methods, which indicates the accuracy of a method for community detection and is used to evaluate the quality of each node-to-community assignment. The most important strength of BiVoting is calculating a higher modularity score in large-scale real bipartite networks. Of course, it should be noted that fewer communities are found, and this reduction improves as the network size increases. Fewer communities are detected due to BiVoting's ability to accurately replicate the voting process. Some of the compared community detection methods like AsymIntimacy, LPBRIM,

and AdaptiveBRIM cannot be used in large real bipartite networks like DW and DP because, after many hours of execution, they either fail or receive no specific result. But BiVoting in these networks is implemented and executed in an acceptable time difference, compared to the BiAttractor method. This is another strength of Bivoting. Also, Bivoting executes faster in small real bipartite networks and somewhat faster in medium real networks. However, due to BiVoting's ability to accurately replicate the voting process, fewer communities are detected in these networks. All of these findings support the notion that Bivoting is a suitable technique for community detection in bipartite networks.

It is proposed in the future to provide a method for community detection in bipartite networks using centrality measures that don't require full knowledge of the network topology and can meet the requirements of today's complex social network environment and uses innovative ideas to discover the community structures in the shortest possible time.

Author Contributions: Conceptualization, A.K., A.M., MM.G.S. and H.M.; methodology, A.K., A.M., MM.G.S. and H.M.; software, A.K. and H.M.; validation, A.K. and H.M.; formal analysis, A.K. and H.M.; investigation, A.K., A.M., MM.G.S. and H.M.; resources, A.K., A.M., MM.G.S. and H.M.; data curation, A.K. and H.M.; writing (original draft preparation), A.K. and H.M.; writing (review and editing), A.K., A.M., MM.G.S. and H.M.; visualization, A.K. and H.M.; supervision, A.K., A.M., MM.G.S. and H.M.; project administration, A.K., A.M., MM.G.S. and H.M. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: Not applicable.

Acknowledgments: I am very grateful to Fatemeh Sarshar Tehrani. She has endured a lot of effort in the preparation of this paper. This paper is taken from the Ph.D Thesis of Ali Khosrozadeh entitled "A Novel Method for Community Detection in Bipartite Networks" at Islamic Azad University, Qazvin branch

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

References

1. Tang, LY.; Li, SN.; Lin, J.G.Q.L.J. Community Structure Detection Based on the Neighbor Node Degree Information. *Int. J. Mod. Phys. C* **2016**, *27*. <https://doi.org/10.1142/s0129183116500467>.
2. Li, W.; Huang, C.W.M.C.X. Stepping Community Detection Algorithm Based on Label Propagation and Similarity. *Phys. A Stat. Mech. Its Appl.* **2017**, *472*, 145–155. <https://doi.org/10.1016/j.physa.2017.01.030>.
3. Yang, JX.; Zhang, X. Finding Overlapping Communities Using Seed Set. *Phys. A Stat. Mech. Its Appl.* **2017**, *467*, 96–106. <https://doi.org/10.1016/j.physa.2016.10.006>.
4. Bilal, S.; Abdelouahab, M. Evolutionary Algorithm and Modularity for Detecting Communities in Networks. *Phys. A Stat. Mech. Its Appl.* **2017**, *473*, 89–96. <https://doi.org/10.1016/j.physa.2017.01.018>.
5. Sun, HL.; Chng, E.Y.X.G.J.S.S. An Improved Game-Theoretic Approach to Uncover Overlapping Communities. *Int. J. Mod. Phys. C* **2017**, *28*. <https://doi.org/10.1142/s0129183117501121>.
6. Evans, TS.; Lambiotte, R. Line Graphs, Link Partitions and Overlapping Communities. *Phys. Rev. E* **2009**, *80*, 016105. <https://doi.org/10.1103/physreve.80.016105>.
7. Blondel, VD.; Guillaume, J.L.R.L.E. Fast Unfolding of Communities in Large Networks. *J. Stat. Mech.: Theory Exp.* **2008**. <https://doi.org/10.1088/1742-5468/2008/10/p10008>.
8. Zhou, T.; Ren, J.M.M.Z.Y. Bipartite Network Projection and Personal Recommendation. *Phys. Rev. E* **2007**, *76*, 046115. <https://doi.org/10.1103/physreve.76.046115>.
9. Wang, X.; Qin, X. Asymmetric Intimacy and Algorithm for Detecting Communities in Bipartite Networks. *Phys. A Stat. Mech. Its Appl.* **2016**, *462*, 569–578. <https://doi.org/10.1016/j.physa.2016.06.096>.
10. Newman, M. Modularity and Community Structure in Networks. *PNAS* **2006**, *103*, 8577–8582. <https://doi.org/10.1073/pnas.0601602103>.
11. Fortunato, S.; Barthelemy, M. Resolution Limit in Community Detection. *PNAS* **2007**, *104*, 36–41. <https://doi.org/10.1073/pnas.0605965104>.
12. Lehmann, S.; Schwartz, M.H.L. Biclique Communities. *Phys. Rev. E* **2008**, *78*, 016108. <https://doi.org/10.1103/physreve.78.016108>.
13. Newman, M. Finding Community Structure in Networks Using the Eigenvectors of Matrices. *Phys. Rev. E* **2006**, *74*, 036104. <https://doi.org/10.1103/physreve.74.036104>.

14. Guimera, R.; Sales-Pardo, M.A.L. Module Identification in Bipartite and Directed Networks. *Phys. Rev. E* **2007**, *76*, 036102. <https://doi.org/10.1103/physreve.76.036102>.
15. Zhang, P.; Wang, J.L.X.L.M.D.Z.F.Y. Clustering Coefficient and Community Structure of Bipartite Networks. *Phys. A Stat. Mech. Its Appl.* **2008**, *387*, 6869–6875. <https://doi.org/10.1016/j.physa.2008.09.006>.
16. Cui, Y.; Wang, W. Uncovering Overlapping Community Structures by the Key Bi-Community and Intimate Degree in Bipartite Networks. *Phys. A Stat. Mech. Its Appl.* **2014**, *407*, 7–14. <https://doi.org/10.1016/j.physa.2014.03.077>.
17. Larremore, D.B.; Clauset, A.J.A. Efficiently Inferring Community Structure in Bipartite Networks. *Phys. Rev. E* **2014**, *90*, 012805. <https://doi.org/10.1103/physreve.90.012805>.
18. Xu, Y.; Chen, L.L.B.L.W. Density-Based Modularity for Evaluating Community Structure in Bipartite Networks. *Inf. Sci.* **2015**, *317*, 278–294. <https://doi.org/10.1016/j.ins.2015.04.049>.
19. Liu, J.G.; Hou, L.P.X.G.Q.Z.T. Stability of Similarity Measurements for Bipartite Networks. *Sci. Rep.* **2016**, *6*. <https://doi.org/10.1038/srep18653>.
20. Sun, H.L.; Chng, E.Y.X.G.J.S.S.C.D. A Fast Community Detection Method in Bipartite Networks by Distance Dynamics. *Phys. A Stat. Mech. Its Appl.* **2018**, *496*, 108–120. <https://doi.org/10.1016/j.physa.2017.12.099>.
21. Newman, M.E.; Girvan, M. Finding and Evaluating Community Structure in Networks. *Phys. Rev. E* **2004**, *69*, 026113. <https://doi.org/10.1103/physreve.69.026113>.
22. Zhou, C.; Feng, L.Z.Q. A Novel Community Detection Method in Bipartite Networks. *Phys. A Stat. Mech. Its Appl.* **2018**, *492*, 1679–1693. <https://doi.org/10.1016/j.physa.2017.11.089>.
23. Al-sharoa, E.; Aviyente, S. A Unified Spectral Clustering Approach for Detecting Community Structure in Multilayer Networks. *Symmetry* **2023**, *15*, 1368. <https://doi.org/10.3390/sym15071368>.
24. Al-sharoa, E.; Al-wardat, M.A.k.M.A.B.A. Discovering Community Structure in Multiplex Networks via a Co-Regularized Robust Tensor-Based Spectral Approach. *Appl. Sci.* **2023**, *13*, 2514. <https://doi.org/10.3390/app13042514>.
25. Anderson, A.; Potikas, P.P.K. CoDiS: Community Detection via Distributed Seed Set Expansion on Graph Streams. *Information* **2023**, *14*, 594. <https://doi.org/10.3390/info14110594>.
26. Cai, J.; Li, W.Z.X.W.J. New Random Walk Algorithm Based on Different Seed Nodes for Community Detection. *Mathematics* **2024**, *12*, 2374. <https://doi.org/10.3390/math12152374>.
27. Chen, R.; Liang, B. Research on the Prediction of Operator Users Number Portability Based on Community Detection. *Appl. Sci.* **2023**, *13*, 3497. <https://doi.org/10.3390/app13063497>.
28. Chen, Y.; Ye, W.L.D. Spectral Clustering Community Detection Algorithm Based on Point-Wise Mutual Information Graph Kernel. *Entropy* **2023**, *25*, 1617. <https://doi.org/10.3390/e25121617>.
29. Chen, D.; Nie, M.X.F.W.D.C.H. Link Prediction and Graph Structure Estimation for Community Detection. *Mathematics* **2024**, *12*, 1269. <https://doi.org/10.3390/math12081269>.
30. Christopoulos, K.; Baltso, G.T.K. Local Community Detection in Graph Streams with Anchors. *Information* **2023**, *14*, 332. <https://doi.org/10.3390/info14060332>.
31. Choudhury, N. Community-Aware Evolution Similarity for Link Prediction in Dynamic Social Networks. *Mathematics* **2024**, *12*, 285. <https://doi.org/10.3390/math12020285>.
32. Ferdowsi, A.; Dehghan Chenary, M. Gain and Pain in Graph Partitioning: Finding Accurate Communities in Complex Networks. *Algorithms* **2024**, *17*, 226. <https://doi.org/10.3390/a17060226>.
33. Gao, K.; Ren, X.Z.L.Z.J. Automatic Detection of Multilevel Communities: Scalable, Selective and Resolution-Limit-Free. *Appl. Sci.* **2023**, *13*, 1774. <https://doi.org/10.3390/app13031774>.
34. Huang, J.; Gu, Y. Unsupervised Community Detection Algorithm with Stochastic Competitive Learning Incorporating Local Node Similarity. *Appl. Sci.* **2023**, *13*, 10496. <https://doi.org/10.3390/app131810496>.
35. Huang, Z.; Xu, W.Z.X. Community-CL: An Enhanced Community Detection Algorithm Based on Contrastive Learning. *Entropy* **2023**, *25*, 864. <https://doi.org/10.3390/e25060864>.
36. Huang, C.; Zhong, Y. An Algorithm Based on Non-Negative Matrix Factorization for Detecting Communities in Networks. *Mathematics* **2024**, *12*, 619. <https://doi.org/10.3390/math12040619>.
37. K., H.; V., J.P.K.P.E. Distributed Genetic Algorithm for Community Detection in Large Graphs with a Parallel Fuzzy Cognitive Map for Focal Node Identification. *Appl. Sci.* **2023**, *13*, 8735. <https://doi.org/10.3390/app13158735>.
38. Kamuhanda, D.; Cui, M.T.C. Illegal Community Detection in BitCoin Transaction Networks. *Entropy* **2023**, *25*, 1069. <https://doi.org/10.3390/e25071069>.
39. Lee, S.; Lee, J.L.J.C.H.Y.J. A Network Analysis Approach to Detecting Social Issues with Web-Based Data. *Appl. Sci.* **2023**, *13*, 8516. <https://doi.org/10.3390/app13148516>.

40. Liu, Y.; Liu, W.T.X.Y.H.Y.P.X.X.W.Y. CSIM: A Fast Community Detection Algorithm Based on Structure Information Maximization. *Electronics* **2024**, *13*, 1119. <https://doi.org/10.3390/electronics13061119>.
41. Liu, C.; Han, Y.X.H.Y.S.W.K.S.Y. A Community Detection and Graph-Neural-Network-Based Link Prediction Approach for Scientific Literature. *Mathematics* **2024**, *12*, 369. <https://doi.org/10.3390/math12030369>.
42. Lubashevskiy, V.; Ozaydin, S.O.F. Improved Link Entropy with Dynamic Community Number Detection for Quantifying Significance of Edges in Complex Social Networks. *Entropy* **2023**, *25*, 365. <https://doi.org/10.3390/e25020365>.
43. Mai, T.; Crane, M.B.M. Students Learning Behaviour in Programming Education Analysis: Insights from Entropy and Community Detection. *Entropy* **2023**, *25*, 1225. <https://doi.org/10.3390/e25081225>.
44. Papei, H.; Li, Y. Stochastic Local Community Detection in Networks. *Algorithms* **2023**, *16*, 22. <https://doi.org/10.3390/a16010022>.
45. Qing, H. Estimating Mixed Memberships in Directed Networks by Spectral Clustering. *Entropy* **2023**, *25*, 345. <https://doi.org/10.3390/e25020345>.
46. Qing, H. Estimating the Number of Communities in Weighted Networks. *Entropy* **2023**, *25*, 551. <https://doi.org/10.3390/e25040551>.
47. Sachpenderis, N.; Koloniari, G. Outlier Detection and Prediction in Evolving Communities. *Appl. Sci.* **2024**, *14*, 2356. <https://doi.org/10.3390/app14062356>.
48. Sanchez-Moreno, D.; Lopez Batista, V.M.V.M.S.L.A.M.G.M. Social Network Community Detection to Deal with Gray-Sheep and Cold-Start Problems in Music Recommender Systems. *Information* **2024**, *15*, 138. <https://doi.org/10.3390/info15030138>.
49. Sawicki, J.; Ganzha, M.P.M. Reddit Crosspostnet: Studying Reddit Communities with Large-Scale Crosspost Graph Networks. *Algorithms* **2023**, *16*, 424. <https://doi.org/10.3390/a16090424>.
50. Sawicki, J.; Ganzha, M. Exploring Reddit Community Structure: Bridges, Gateways and Highways. *Electronics* **2024**, *13*, 1935. <https://doi.org/10.3390/electronics13101935>.
51. Song, A.; Ji, R.Q.W.Z.C. RGCLN: Relational Graph Convolutional Ladder-Shaped Networks for Signed Network Clustering. *Appl. Sci.* **2023**, *13*, 1367. <https://doi.org/10.3390/app13031367>.
52. Song, Y.; Zheng, Z.S.Y.W.B. GLOD: The Local Greedy Expansion Method for Overlapping Community Detection in Dynamic Provenance Networks. *Mathematics* **2023**, *11*, 3284. <https://doi.org/10.3390/math11153284>.
53. Stein, J.; Ott, D.N.J.B.D.S.M.F.S. NISQ-Ready Community Detection Based on Separation-Node Identification. *Mathematics* **2023**, *11*, 3323. <https://doi.org/10.3390/math11153323>.
54. Tang, F.; Zhao, X.L.C. Community Detection in Multilayer Networks Based on Matrix Factorization and Spectral Embedding Method. *Mathematics* **2023**, *11*, 1573. <https://doi.org/10.3390/math11071573>.
55. Wang, F.; Hu, F.C.R.X.N. HLEGF: An Effective Hyper Network Community Detection Algorithm Based on Local Expansion and Global Fusion. *Mathematics* **2023**, *11*, 3497. <https://doi.org/10.3390/math11163497>.
56. Xia, X.; Ma, W.Z.J.Z.E. Community-Enhanced Contrastive Learning for Graph Collaborative Filtering. *Electronics* **2023**, *12*, 4831. <https://doi.org/10.3390/electronics12234831>.
57. Xu, H.; Ran, Y.X.J.T.L. An Influence-Based Label Propagation Algorithm for Overlapping Community Detection. *Mathematics* **2023**, *11*, 2133. <https://doi.org/10.3390/math11092133>.
58. Yaakob, A.; Shafie, S.G.A.R.S.K.K. Large-Scale Group Decision-Making Method Using Hesitant Fuzzy Rule-Based Network for Asset Allocation. *Information* **2023**, *14*, 588. <https://doi.org/10.3390/info14110588>.
59. Yang, Y.; Yu, S.P.B.L.C.L.M. Community Detection in Multiplex Networks Using Orthogonal Non-Negative Matrix Tri-Factorization based on Graph Regularization and Diversity. *Mathematics* **2024**, *12*, 1124. <https://doi.org/10.3390/math12081124>.
60. Yao, B.; Zhu, J.M.P.G.K.R.X. A Constrained Louvain Algorithm with a Novel Modularity. *Appl. Sci.* **2023**, *13*, 4045. <https://doi.org/10.3390/app13064045>.
61. Yow, J.; Liu, B. Community-Detection Method of Complex Network Based on Node Influence Analysis. *Symmetry* **2024**, *16*, 754. <https://doi.org/10.3390/sym16060754>.
62. Yu, L.; Guo, X.Z.D.Z.J. A Multi-Objective Pigeon-Inspired Optimization Algorithm for Community Detection in Complex Networks. *Mathematics* **2024**, *12*, 1486. <https://doi.org/10.3390/math12101486>.
63. Zalik, K.; Zalik, M. Density-Based Entropy Centrality for Community Detection in Complex Networks. *Entropy* **2023**, *25*, 1196. <https://doi.org/10.3390/e25081196>.
64. Zhang, W.; Yu, S.W.L.G.W.L.M. Constrained Symmetric Non-Negative Matrix Factorization with Deep Auto-Encoders for Community Detection. *Mathematics* **2024**, *12*, 1554. <https://doi.org/10.3390/math12101554>.

65. Zhao, C.; Al-Bashabsheh, A.C.C. Game Theoretic Clustering for Finding Strong Communities. *Entropy* **2024**, *26*, 268. <https://doi.org/10.3390/e26030268>.
66. Mahyar, H. Detection of Top-k Central Nodes in Social Networks: A Compressive Sensing Approach. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* **2015**, pp. 902–909. <https://doi.org/10.1145/2808797.2808811>.
67. Mahyar, H.; Rabiee, H.M.A.G.E.N.A. CS-Comdet: A Compressive Sensing Approach for Intercommunity Detection in Social Networks. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* **2015**, pp. 89–96. <https://doi.org/10.1145/2808797.2808856>.
68. Opsahl, T. Triadic Closure in Two-Mode Networks: Redefining the Global and Local Clustering Coefficients. *Soc. Netw.* **2013**, *35*, 159–167. <https://doi.org/10.1016/j.socnet.2011.07.001>.
69. Taheri, SM.; Mahyar, H. Hellrank: A Hellinger-Based Centrality Measure for Bipartite Social Networks. *Soc. Netw. Anal. Min.* **2017**, *7*. <https://doi.org/10.1007/s13278-017-0440-7>.
70. Barber, M. Modularity and Community Detection in Bipartite Networks. *Phys. Rev. E* **2007**, *76*, 066102. <https://doi.org/10.1103/physreve.76.066102>.
71. Liu, X.; Murata, T. Community Detection in Large-Scale Bipartite Networks. *J-STAGE* **2010**, *25*, 50–57. <https://doi.org/10.1527/tjsai.25.16>.
72. Kunegis, J. Konect: The Koblenz Network Collection. *Proceedings of the 22th International Conference on World Wide Web* **2013**, p. 1343–1350. <https://doi.org/10.1145/2487788.2488173>.
73. Nacher, JC.; Schwartz, J. Modularity in Protein Complex and Drug Interactions Reveals New Polypharmacological Properties. *PLOS ONE* **2012**, *7*, 1–13. <https://doi.org/10.1371/journal.pone.0030028>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.