Community Detection Algorithm Based on High Degree Node Selection in Complex Networks

[1]Amedapu Srinivas and [2]Leela Velusamy
[1,2]Department of Computer Science & Engineering, NIT Trichy, National Institute of Technology, Tamil Nadu, India.
[1]srinivasreddyamedapu@yahoo.com, [2]leela@nitt.edu.in

Correspondence should be addressed to Amedapu Srinivas : srinivasreddyamedapu@yahoo.com

ArticleInfo
Journal of Machine and Computing (https://anapub.co.ke/journals/jmc/jmc.html)
Doi : https://doi.org/10.53759/7669/jmc202505146.
Received 22 March 2025; Revised from 30 May 2025; Accepted 17 June 2025.
Available online 05 July 2025.

Abstract – Community detection plays a central role in the analysis of social networks, where individuals naturally form structured groups such as neighborhood clusters or small rural communities. A key challenge in this domain is accurately identifying these communities—commonly defined as subsets of nodes that are more densely connected internally than with the rest of the network. Traditional methods often rely on hierarchical clustering for this task. However, recent research has explored alternative approaches involving various clustering strategies and connectivity-based evaluation metrics. In this study, we introduce a novel method called the Biggest Degree Head Node Technique (BDNHT) and evaluate its effectiveness against the conventional Random Head Node Technique. The proposed method focuses on selecting an optimal set of centroids using fitness-based criteria, aiming to achieve more meaningful and well-separated community structures.

Keywords – Network Node, Clustering, Biggest Degree Head, Social Networks, Longest Distance Head, Social Community, Cluster.

I. INTRODUCTION

Complex network analysis is now a central pillar in the analysis of modern data analysis, and researchers can now model, visualize, and interpret complex associations in various systems in the real world [1]. Social networks of biological interaction maps and collaboration graphs, transportation networks, and the organization of the World Wide Web are all examples of powerful abstractions to represent pairwise relationships between objects [2]. Social networks, in particular, those created through platforms like Facebook [3], Twitter [4], YouTube [5], and Wikipedia [6], have enjoyed special attention among being manifold types of networks on account of their large size, abundance of user-generated content, and dynamics. Community detection is one of the most significant and problematic projects of the analysis of such networks, to reveal clandestine group structures in a network [7]. Say a network has a set of nodes that have a denser connection between themselves than they do to the rest of the network; then such a set of nodes is usually considered a community within the network. The identification of such communities is important to a plethora of applications: Companies can do targeted marketing, platforms can construct personalized recommendation systems, work on fraud detection, understand social dynamics, propagate information, and identification of influencers. The nature of social media networks generates complex and large-scale graphs that are sparse, noisy, heterogeneous, and overlapping community structures in nature [8]. There are more traditional community algorithms, including splitable clustering, such as hierarchical clustering [9], spectral clustering [10], and modularity optimization [11] (e.g., Girvan-Newman algorithm [12]), which have shown success in dealing with medium-sized datasets where communities have a clear structure. The methods, however, face problems with scaling to, adapting to, and interpreting real-world community structures in social networks, where communities can be highly diverse in community size, density, and community definability. Moreover, most of the methods call upon the previous knowledge of the number of communities, and this assumption is seldom fulfilled in practice.

In response to these shortcomings, in order to emerge with a community detection strategy that is computationally effective, this research develops a new yet computationally effective community detection strategy named the Biggest Degree Node Head Technique (BDHNT). The point is that the topological core of nodes in a network, namely degree centrality, can be used as the default starting point of community formation that is more natural, intuitive, and which may easily translate in real life. Nodes having maximum degree are chosen as head nodes or community centroids under the premise that high-degree nodes have more chances of being influential and well-networked representatives of underlying communities. After getting the results of the head nodes, the algorithm continues assigning the rest of the nodes to the community by using the shortest path distance of each node to the nearest centroid. The distance-based assignment approach will guarantee that nodes are clustered with structurally nearby leaders, hence achieving a better intra-community bond as well as inter-community distance. To further refine the quality of discovered communities, the method includes a fitness-based optimization procedure that iteratively ranks and refines community memberships to optimize measures of cluster quality like density, modularity, and separability. Contrary to most available approaches, BDHNT does not involve prior knowledge of the number of communities, and hence it can be used for unsupervised and exploratory analysis. Having a low computational cost, being interpretable in its design, and being deterministic, it is particularly appealing to large-scale network mining tasks that require transparency and scalability.

The motivation of the work assumes a more frequent requirement of lightweight, interpretable, and flexible algorithms that can run efficiently on large, noisy, and dynamically changing networks without requiring a large number of parameters to tune or seed the outcome with randomization. Basing community detection on the humble and mighty foundations of graph-theoretic concepts, BDHNT offers a new way to look at scalable clustering in social networks. A wide range of experiments is carried out to root the efficiency of the suggested approach on the Wikipedia Vote Network data, in which a popularity fight among Wikipedia users consists of the voting graph across the real-world graph that shows the voting conduct of Wikipedia users in the case of an election of administrators.

Our key contributions to this research include:

- We suggested BDHNT, a new community detection algorithm, which chooses nodes with a high degree in nature as cluster centroids so that an efficient and understandable community can be constructed with them.
- We proposed a deterministic and unsupervised clustering algorithm that does not use random initialization, and the number of communities does not have to be known in advance.
- We added an iterative refinement measure based on concepts of fitness to reinforce the structural integrity of the detected communities in terms of significant measures of the graph related to random graphs, e.g., density and separability.
- We presented a light and scalable method that is suitable for real-life applications in mining social networks, marketing intelligence, detecting anomalies, or recommendations.

II. RELATED CONCEPTS

Many approaches to community detection algorithms have been proven over the years. Each trend is efficient and effective in their way. Zhao, Liang and Wang [13] suggested a new community detection algorithm using graph compression in order to enhance efficiency in large-scale social networks. They do it by iteratively combining low-degree vertices and finding community seeds based on the density and quality index of the vertex. Its communities are expanded and projected back into the original network. Through experimental results, it is indicated that the method is superior in terms of accuracy and scalability compared to the other available state-of-the-art algorithms. Mester et al. [14] proposed the dual perspective of measuring node importance in complex networks where communities are combined with global centrality measures. Their approach emphasizes the fact that these two views, coupled with each other, are offering overlapping as well as complementary ideas about identifying the best influential nodes. This two-fold evaluation proves to be successful in the validation of robustness of networks by using experimental validation on both synthetic and real-world networks, and better knowledge of the complex dynamics of structures.

Li et al. [15] presented a strengthened node representation technique in detecting communities, which was the combination of the global embeddings of communities and the local embeddings of nodes. The approach they have taken into account node influence, community membership, and structural similarity so that they are able to be more expressive. This mixture model of embedding not only increases the performance of node and community representation learning but also has the ability to detect overlapping communities in complex networks. Boroujeni and Soleimani [16] dealt with both problems, namely community discovery and influential node mapping in complex networks. They estimate the influence sphere of crucial nodes to delineate communities and aim to optimise modularity, an NP-hard task, by heuristic techniques. The proposed solution is grounded in the principles of scale-free networks and demonstrates competitive performance on real-world datasets, effectively identifying the most important node within each community. Zhao et al. [17] examined the structure and evolution of scientific research cooperation networks concerning core node ratings, community detection, and layout techniques. Their method accommodates both network topology and node heterogeneity, enhancing community detection and the visualisation of collaborative structures. Their methodology, grounded in the network embedding of research qualities, effectively reveals the underlying structure of scientific collaboration and aids in scientific management and policy development.

Kumar, Panda, and Aggarwal [18] suggested a new choice based on the community detection method supported by network embedding and the gravitational search optimization. They included the nodes of a graph into a vector space, approximated the graph by a low-rank approximation to mitigate noise, and applied the graph nodes' localized k-means clustering by a search algorithm based on gravitational forces. Embedding tests on real and generational networks confirm the performance of their framework to identify the significant architecture of communities. Masooleh et al. [19] suggested a new community detection algorithm, which is an improvement of the Whale Optimization Algorithm (WOA), a multi-objective extension of WOA. They discretize the positions of populations, reformulate initialization and updating features, and sort out Pareto-optimal combinations of communities with the help of non-dominated sorting. Benchmark data experiments, in addition to the Tennessee Eastman process, show the effectiveness and scalability of applying the method to discover community structures. Samie, Behbood, and Hamzeh [20] suggested improving community identification within social networks using the Two-phase Influence Maximization. These options transform a published local community detection algorithm to suit it to detect influential seed nodes more accurately and efficiently. Further, they present a method of dynamical networks that identifies the initial nodes in every snapshot without restarting calculations, which are time-consuming. In both the static and the dynamic conditions, experimental results indicate better performance compared to the conventional techniques.

Al-Andoli, Cheah, and Tan [21] proposed an innovative community discovery system utilising a deep autoencoder, augmented with Particle Swarm Optimisation (PSO) and continuation methods. These techniques assist the model in circumventing local minima and premature convergence problems prevalent in gradient-based training, particularly in extensive networks. Their method efficiently reveals community patterns by concurrently minimising reconstruction loss and maximising modularity. Empirical findings from 11 real-world datasets indicate enhanced performance relative to current deep learning methodologies. They have also suggested an approach [22] to community detection referred to as a deep autoencoder that operates in dealing with the inefficiencies that exist in large networks and resorts to employing the idea of the partitioning of networks and reductions of parameters, and sharing of parameters. Their design involves a parallel design and a new similarity constraint to preserve the detection performance, but to (massively) accelerate training and scaling. Without compromising accuracy, experiments show greater efficiency, particularly at the higher values of partitioning. To detect key nodes in propagandistic communities on social neighborhoods,

Khanday et al. [23] came up with an algorithm named Boundary-based Community Detection Approach (BCDA). They have a two-step approach with which they identify communities with both boundary and interior nodes through the Leader Ranker algorithm and Constraint Coefficient. When applied to a custom Twitter dataset, the model has been effective in identifying six propagandistic communities as well as outperforming the existing approaches to detecting those, in particular, ICRIM or CBIMA, particularly during high-impact events: the COVID-19 pandemic. Aldabobi, Sharieh, and Jabri [24] enhanced the Louvain algorithm (LVA) by incorporating node significance through degree centrality to the community detection task, producing the Improved Louvain Algorithm (ILVA). In doing so, ILVA maximizes modularity and takes advantage of node importance to inform the scanning sequence, producing more consistent and better-quality community structures. The real-world network experiments proved that ILVA is more stable and modular without loss of efficiency. Shang et al. [25] proposed a novel approach for local community detection that alternates between robust and weak fusion strategies to improve node assignment. The robust fusion technique employs an innovative membership function that incorporates both node and connection-based information, whereas the weak fusion finds influential nodes through a parameter-constrained similarity measure. They exhibit superior performance in precision and stability compared to six existing state-of-the-art algorithms, and offer a metric of community fitness that aids in optimising community detection procedures without requiring ground truth.

Overall, the existing methods of community detection have achieved tremendous improvements due to different methods, such as modularity optimization, heuristics based on centrality, graph compression, deep learning machines, and fusion techniques. These techniques are quite instructive, but they tend to fall short of the goal in the presence of large-scale or dynamic networks, because they tend to meet issues of high computation time or inability to scale up, sensitivity to parameter settings, and the inability to provide interpretations. Inspired by these constraints, the following research introduces the BDHNT, an attempt to provide a computationally efficient, scalable, and interpretable method of community identification. BDHNT eliminates essential weaknesses of models that came earlier and is successful at being simple and having practical application through the use of degree centrality to select the initial community heads and a distance-based node assignment mechanism, which are then improved by the use of individual fitness. In the remaining sections, the description of the methodology, as well as an assessment of the performance of the proposed approach compared with available state-of-the-art algorithms, will be presented.

III. METHODOLOGY

The general purpose of community detection is to divide a network so that the nodes of one group (or community) relate more with nodes of the same group than with nodes of another group. Manuscript: Developing a new algorithm in this work, the BDHNT can improve community detection because of its ability to accommodate a more precise centroid definition in terms of local structural characteristics. The given approach is compared with another one, the Random Head Node Technique (RHNT). The two methods are based on the same idea of assigning nodes according to the shortest paths, but differ in terms of how the centroid is chosen and refined. **Fig 1** shows the conceptual workflow of the suggested BDHNT approach that indicates the main stages of the work, beginning with the centroid selection to the iterative refinement and convergence.

Random Node Head Technique (RNHT)

Random Head Node Technique (RHNT) is a fundamental community detection network that works upon the rule of proximity of the shortest path to a randomly selected centroid. The algorithm starts by choosing k random nodes of the network that will act as a centroid or head of the community. After the selection, every node in the network is made sure to belong to the community of the centroid, in which it has the shortest hop distance. The formal description of this assignment can be as follows:

$$Assign(v) = \arg\min_{c_i \in C} d(v, c_i) \tag{1}$$

where v is a point in the current network, $C = \{c_1, c_2, \ldots, c_k\}$ is the set of the centroid points, and $d\,(v, ci)$ is the shortest path as measured between point v and ci in the centroid set. **Fig 2** presents a graphical example of the structure that is used to demonstrate this technique.

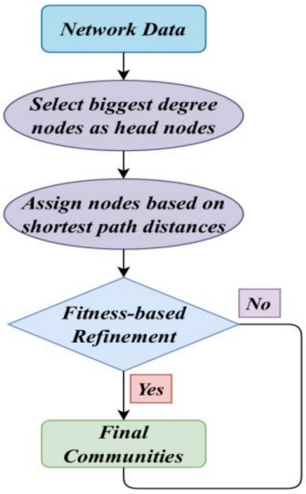

Fig 1. Workflow of the Proposed Biggest Degree Node Head Technique (BDHNT) For Community Detection.

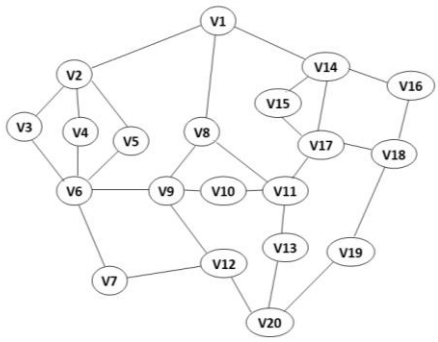

Fig 2. Sample Connected Graph.

Once all the nodes have been labeled, the fitness of the community structure so created is then measured in relation to the inter-community links. Because the centroids are randomly selected, this is performed several times with unrelated sets of centroids, and the configuration that reaches the lowest fitness value is chosen as the final clustering. Nevertheless, even though its simplicity grants RHNT stability, the setting of an arbitrary centroid leads to sub-optimal clusters.

Biggest Degree Node Head Technique (BDHNT)

Dimensionality of the major degree nodes head technique BDHNT. To overcome the drawbacks of RHNT, a structure-based approach to centroid improvement is suggested as part of the proposed BDHNT. It also starts with an initial set of randomly chosen centroids and allocates the nodes to their closest centroid according to the shortest path distance. But in further iterations, the centroids are not chosen randomly. Rather, a node with maximum degree is picked as the new centroid per community.

The degree of a node v, which in other words is the number of direct links that node has with other nodes, is stated as:

$$deg(v) = |\{u \in V : (v, u) \in E\}| \tag{2}$$

Let $C_i^{(t)}$ be the ith cluster at iteration t. The new centroid of cluster i in the next iteration is obtained as:

$$c_i^{(t+1)} = \arg\max_{v \in C_i^{(t)}} (v) \tag{3}$$

Once new centroids are chosen, the distances between all nodes and the centroids are recalculated, and nodes are redistributed to the closest centroid. Such updating of the centroids according to the degree and the re-assignment of nodes is repeated until no substantial changes are made with the algorithm converging on a final community structure. The rationale of such an approach is that high-degree nodes tend to be central in their local communities and, therefore, more valuable anchors of clustering.

Community Fitness Evaluation

A fitness function is used to compare and measure the quality of a community at different iterations in a quantitative fashion. The meaning of this function is the statistical average number of external links at each node in every cluster. An external link is defined as the relation between a node and any out-of-community node. The fitness of a particular clustering at the t-th iteration, which will be denoted as $F^{(t)}$, is as follows:

$$F^{(t)} = \sum_{i=1}^{k} \left(\frac{1}{n_i^{(t)}} \sum_{j=1}^{n_i^{(t)}} ext\left(v_{ij}^{(t)}\right) \right) \tag{4}$$

where the variable $n_i^{(t)}$ represents the number of nodes in the cluster i at iteration t, $v_{ij}^{(t)}$ is the jth node in the cluster i, $xt\left(v_{ij}^{(t)}\right)$ is the number of connections of the node $v_{ij}^{(t)}$ to the other nodes not in the cluster i.

The main goal of the algorithm is a reduction of $F^{(t)}$. The lower the measure of fitness, the higher the internal cohesion among communities and the lower the external bonds, so the higher the quality of clustering. This fitness depends neither on ground-truth labels nor can it be calculated only in supervised environments, which makes it a very versatile means of estimating community detection methods.

Iterative Refinement and Convergence

Iteration is the key factor in the BDHNT performance. The community structure gets finer and finer with every recalculation of nodes to a new value of the centroid. Centroids are recalculated in each iteration, choosing the node with the highest degree within each community. The communities are then rebuilt on the basis of minimized path distance to these new centroids.

This is done until one of the following two occurrences happens.

Centroid Stability

When the centroids remain the same in two consecutive iterations:

$$C^{(t+1)} = C^{(t)} \tag{5}$$

Maximum Iterations Reached

When the iterations t become larger than a set similarity threshold $T_{\max}$:

$$t \geq T_{max} \Longrightarrow terminate \tag{6}$$

Algorithm 1. Algorithm of the Biggest Degree Node Head Technique

Input: Network of nodes
Output: Clustered groups of similar nodes
1. **Start**
2. Randomly choose *n* nodes from the network to act as initial centroids.
3. **Repeat until convergence or maximum iterations:**
    3.1. For each centroid:
        3.1.1. Calculate the shortest path between the centroid and every other node.
    3.2. Assign each node to the nearest centroid based on the shortest path to form clusters.
    3.3. For each cluster:
        3.3.1. Compute a fitness score to evaluate cluster quality.
    3.4. For each node in the cluster:
        3.4.1. Count the number of direct connections (degree of the node).
    3.5. Select the node with the highest number of connections in each cluster as the new centroid.
4. After final iteration, identify the set of centroids that produced the best overall fitness.
5. Generate the final clusters based on those optimal centroids.
6. **End**

Algorithm 1 presents the outline of all the main steps of the iterative refinement process in BDHNT rewards: it begins with the initialization of the centroid and continues with convergence. The most feasible clustering solution is then selected after the convergence as the one scoring the lowest fitness value in all the iterations:

$$F_{best} = \min_{t \in [1, T_{max}]} F^{(t)} \tag{7}$$

This makes sure the algorithm not only ceases effectively but also holds on to the most excellent partitioning it will have accomplished as it functions. The advantage of BDHNT is that, whereas topological centrality (e.g., node degrees) can be exploited, proximity (e.g., shortest paths) is equally used, thus generating a community with a high structural meaning and a compact size. Its structure also meets the standard of interpretability, scalability, and the possibility to adapt to various forms of complex networks.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this part, the results obtained from the research methodology are explained. In terms of fitness, separability, density, execution time, and memory consumption during execution, the results obtained from the Random Node Head and the Biggest Degree Node Head are compared. The research is carried out using Java (JDK 1.7) on a Windows 7 32-bit machine with a 2.94GHz Core 2 Duo processor and 2 GB of RAM.

Dataset Description

For our experimental study, we utilized data from the Wikipedia Vote Network, accessible at http://snap.stanford.edu/data/wiki-Vote.html. Wikipedia, as a collaborative platform, allows contributors from around the world to edit and maintain its content. Within this community, certain users can be elevated to administrator status through a process known as the Request for Adminship (RfA), where fellow contributors participate in public discussions and voting to determine the outcome.

The dataset originates from a comprehensive dump of Wikipedia's edit history dated January 3, 2008, and captures the full spectrum of adminship-related voting activity up to that point. It comprises 2,794 election events involving 103,663 votes cast by 7,066 users, who either stood for adminship or voted in the process. Out of these elections, 1,235 resulted in successful promotions, while 1,559 were unsuccessful. The voting behavior reflects a nearly equal distribution between regular users and existing administrators. This network effectively maps the voting interactions and community dynamics throughout Wikipedia's early history.

Evaluation Metrics

Separability refers to how distinctly a community is isolated from the remainder of the network. A well-defined community should have strong internal connectivity and minimal external links. To quantify this, separability is computed as the ratio of the number of internal edges within a given cluster to the number of edges connecting that cluster to the rest of the network. This metric helps evaluate how clearly a cluster is distinguished from others. Let I represent the total number of clusters in the network.

$$Separability = \sum_{i=1}^{I} \frac{no.\ of\ inner\ connection}{no.\ of\ outer\ connection} \tag{8}$$

Density is a measure that reflects how tightly connected the members of a community are. Higher density indicates that the nodes within a cluster have strong interconnections, which is a desirable property for well-formed communities. The density is computed using a specific formula, where CS denotes the size of the cluster, i.e., the number of nodes it contains.

$$Density = \sum_{i=1}^{I} \frac{no.\ of\ inner\ connection}{CS \times (CS - 1)} \tag{9}$$

Section 3.3 explains the fitness calculation, execution time is the total time it takes for each approach to finish the task, memory consumption is the total amount of memory used by each technique during execution.

Performance comparison

Fig 3 presents the fitness comparison across different techniques. When the number of iterations is set to five, the fitness values observed are 4.29 for Random, 3.10 for BDNHT, and 3.15 for LDNHT. Increasing the iterations to ten yields values of 4.11, 3.60, and 3.55 respectively. At twenty-five iterations, the values further reduce to 3.42 (Random), 2.86 (BDNHT), and 2.80 (LDNHT). These results indicate that BDNHT and LDNHT consistently outperform the Random technique, as lower fitness values signify better community structure.

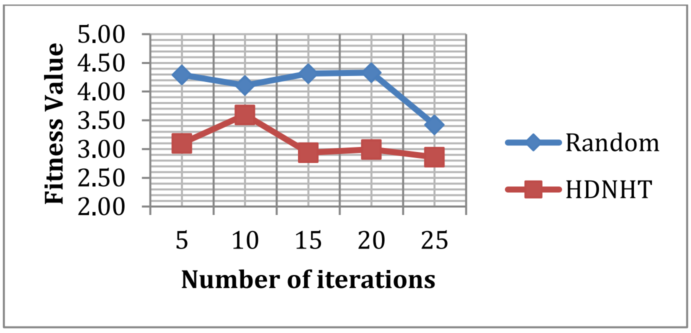

Fig 3. Fitness comparison.

Fig 4 illustrates the comparison of separability across the existing Random technique and the proposed BDNHT and LDNHT methods. When the number of iterations is five, the separability values are 0.94 for Random, 1.23 for BDNHT, and 1.33 for LDNHT. At fifteen iterations, the values increase to 0.95, 1.47, and 1.43 respectively. With twenty-five iterations, the separability further improves to 1.30 (Random), 1.50 (BDNHT), and 1.53 (LDNHT). These results clearly demonstrate that BDNHT and LDNHT outperform the Random approach, as higher separability values indicate more well-defined and distinct communities.

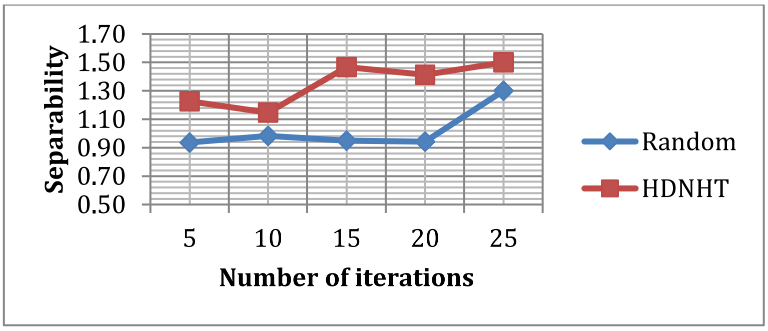

Fig 4. Separability Comparison.

Fig 5 presents a comparison of density between the Random Head Node technique and the proposed Biggest Degree Node Head Technique (BDNHT). At five iterations, the density values are 0.06 for Random and 0.09 for BDNHT. When the number of iterations increases to twenty-five, the values rise to 0.14 and 0.17, respectively. These results indicate that the BDNHT method consistently achieves higher density, which reflects stronger intra-community connections—an essential characteristic of well-formed communities.

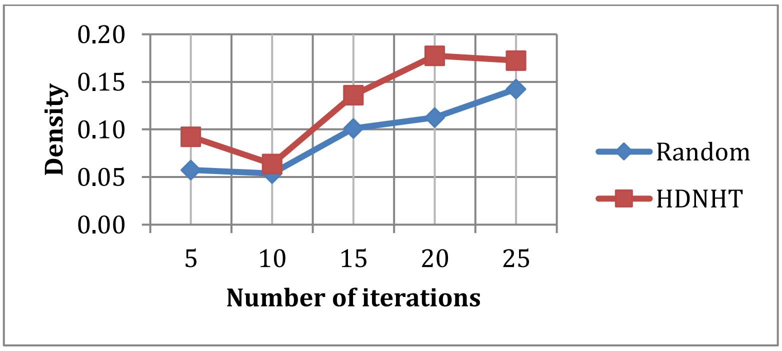

Figure 5: Density Comparison.

Fig 6 compares the execution time of the existing Random Head Node technique and the proposed Biggest Degree Node Head Technique (BDNHT). The results indicate that BDNHT requires approximately 15% more time than the Random approach. Specifically, at five iterations, the execution times are 11,614 ms for Random and 13,218 ms for BDNHT. When the number of iterations increases to twenty-five, the times are 16,014 ms and 17,143 ms, respectively.

While BDNHT consumes slightly more time, the trade-off is justified by its improved performance in terms of community quality.

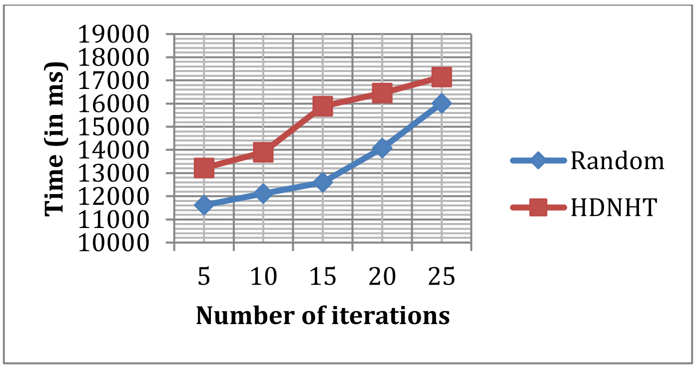

Fig 6 Comparison of Time Taken for Execution.

Fig 7 illustrates the memory usage comparison between the Random Head Node technique and the proposed Biggest Degree Node Head Technique (BDNHT). At five iterations, the memory consumption is 175.38 MB for Random and 175.74 MB for BDNHT. When increased to twenty-five iterations, the values rise to 190.38 MB and 192.54 MB, respectively. These results indicate that the proposed BDNHT method consumes approximately 3% more memory, which is a modest increase considering the performance benefits it offers.

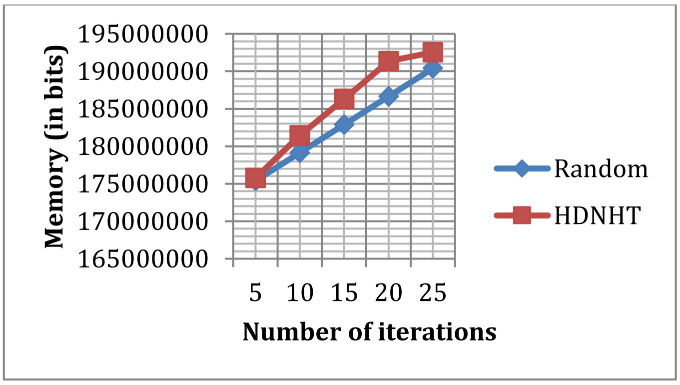

Fig 7. Comparison of Memory Consumption to Execute the Process.

V. CONCLUSION

In this study, we introduce a novel community detection method termed the Biggest Degree Node Head (BDNH) approach for effectively grouping nodes within a network. The method begins by randomly selecting an initial set of centroids, which serve as the basis for the first round of clustering. In each subsequent iteration, these centroids are updated by selecting the node with the highest degree from each cluster formed in the previous step. This ensures that the most connected nodes guide the formation of communities, promoting stronger internal cohesion. To determine the most effective centroid configuration, a fitness value is calculated at each iteration, helping identify the optimal clustering outcome. The performance of the BDNH technique is benchmarked against a baseline method, the Random Head Node Technique, using the Wikipedia Vote Network as the test dataset. A comparative analysis was conducted based on several performance metrics: fitness, separability, density, execution time, and memory usage. The results show that the BDNH technique outperforms the baseline in terms of density and separability, indicating the formation of more cohesive and well-separated communities. However, the Random Head Node Technique demonstrated marginally better results in execution time and memory efficiency, due to its simpler and less computationally intensive nature.

CRediT Author Statement

The authors confirm contribution to the paper as follows:

Conceptualization: Amedapu Srinivas and Leela Velusamy; **Methodology:** Amedapu Srinivas; **Software:** Leela Velusamy; **Data Curation:** Amedapu Srinivas; **Writing- Original Draft Preparation:** Amedapu Srinivas and Leela Velusamy; **Visualization:** Amedapu Srinivas; **Investigation:** Leela Velusamy; **Supervision:** Amedapu Srinivas; **Validation:** Leela Velusamy; **Writing- Reviewing and Editing:** Amedapu Srinivas and Leela Velusamy; All authors reviewed the results and approved the final version of the manuscript.

Data Availability
No data was used to support this study.

Conflicts of Interests
The authors declare no conflict of interest.

Funding
No funding agency is associated with this research.

Competing Interests
There are no competing interests.

References
[1]. B. Kamiński, P. Prałat, and F. Théberge, “Mining Complex Networks,” Nov. 2021, doi: 10.1201/9781003218869.
[2]. S. S. Hussein and A. A. Farhan, “On Linear Algebraic and Graph Theoretic Methods,” Journal of Global Scientific Research, vol. 8, no. 11, pp. 3319–3326, 2023.
[3]. C. He, X. Fei, Q. Cheng, H. Li, Z. Hu, and Y. Tang, “A Survey of Community Detection in Complex Networks Using Nonnegative Matrix Factorization,” IEEE Transactions on Computational Social Systems, vol. 9, no. 2, pp. 440–457, Apr. 2022, doi: 10.1109/tcss.2021.3114419.
[4]. F. Gasparetti, G. Sansonetti, and A. Micarelli, “Community detection in social recommender systems: a survey,” Applied Intelligence, vol. 51, no. 6, pp. 3975–3995, Jun. 2021, doi: 10.1007/S10489-020-01962-3/METRICS.
[5]. S. Rani and M. Kumar, “Ranking community detection algorithms for complex social networks using multilayer network design approach,” International Journal of Web Information Systems, vol. 18, no. 5/6, pp. 310–341, Aug. 2022, doi: 10.1108/ijwis-02-2022-0040.
[6]. Xiaokaiti, Y. Qian, and J. Wu, “Efficient Data Transmission for Community Detection Algorithm Based on Node Similarity in Opportunistic Social Networks,” Complexity, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/9928771.
[7]. S. Mittal, D. Sengupta, and T. Chakraborty, “Hide and Seek: Outwitting Community Detection Algorithms,” IEEE Transactions on Computational Social Systems, vol. 8, no. 4, pp. 799–808, Aug. 2021, doi: 10.1109/tcss.2021.3062711.
[8]. S. Souravlas, S. Anastasiadou, and S. Katsavounis, “A Survey on the Recent Advances of Deep Community Detection,” Applied Sciences, vol. 11, no. 16, p. 7179, Aug. 2021, doi: 10.3390/app11167179.
[9]. T. Li et al., “Hierarchical Community Detection by Recursive Partitioning,” Journal of the American Statistical Association, vol. 117, no. 538, pp. 951–968, Nov. 2020, doi: 10.1080/01621459.2020.1833888.
[10]. K. Berahmand, M. Mohammadi, A. Faroughi, and R. P. Mohammadiani, “A novel method of spectral clustering in attributed networks by constructing parameter-free affinity matrix,” Cluster Computing, vol. 25, no. 2, pp. 869–888, Nov. 2021, doi: 10.1007/s10586-021-03430-0.
[11]. F. Öztemiz and A. Karcı, “KO: Modularity optimization in community detection,” Neural Computing and Applications, vol. 35, no. 15, pp. 11073–11087, Jan. 2023, doi: 10.1007/s00521-023-08284-8.
[12]. R. Kiruthika and M. S. Vijaya, “Community Detection Using Girvan–Newman and Kernighan–Lin Bipartition Algorithms,” Data Intelligence and Cognitive Informatics, pp. 217–231, 2022, doi: 10.1007/978-981-16-6460-1_16.
[13]. X. Zhao, J. Liang, and J. Wang, “A community detection algorithm based on graph compression for large-scale social networks,” Information Sciences, vol. 551, pp. 358–372, Apr. 2021, doi: 10.1016/j.ins.2020.10.057.
[14]. Mester, A. Pop, B.-E.-M. Mursa, H. Greblă, L. Dioşan, and C. Chira, “Network Analysis Based on Important Node Selection and Community Detection,” Mathematics, vol. 9, no. 18, p. 2294, Sep. 2021, doi: 10.3390/math9182294.
[15]. M. Li, S. Lu, L. Zhang, Y. Zhang, and B. Zhang, “A Community Detection Method for Social Network Based on Community Embedding,” IEEE Transactions on Computational Social Systems, vol. 8, no. 2, pp. 308–318, Apr. 2021, doi: 10.1109/tcss.2021.3050397.
[16]. R. Javadpour Boroujeni and S. Soleimani, “The role of influential nodes and their influence domain in community detection: An approximate method for maximizing modularity,” Expert Systems with Applications, vol. 202, p. 117452, Sep. 2022, doi: 10.1016/j.eswa.2022.117452.
[17]. W. Zhao, J. Luo, T. Fan, Y. Ren, and Y. Xia, “Analyzing and visualizing scientific research collaboration network with core node evaluation and community detection based on network embedding,” Pattern Recognition Letters, vol. 144, pp. 54–60, Apr. 2021, doi: 10.1016/j.patrec.2021.01.007.
[18]. S. Kumar, B. S. Panda, and D. Aggarwal, “Community detection in complex networks using network embedding and gravitational search algorithm,” Journal of Intelligent Information Systems, vol. 57, no. 1, pp. 51–72, Nov. 2020, doi: 10.1007/s10844-020-00625-6.
[19]. L. Samandari Masooleh, J. E. Arbogast, W. D. Seider, U. Oktem, and M. Soroush, “An efficient algorithm for community detection in complex weighted networks,” AIChE Journal, vol. 67, no. 7, p. e17205, Jul. 2021, doi: 10.1002/AIC.17205;WGROUP:STRING:PUBLICATION.
[20]. M. E. Samie, E. Behbood, and A. Hamzeh, “Local community detection based on influence maximization in dynamic networks,” Applied Intelligence, vol. 53, no. 15, pp. 18294–18318, Aug. 2023, doi: 10.1007/S10489-022-04403-5/METRICS.
[21]. M. Al-Andoli, W. P. Cheah, and S. C. Tan, “Deep autoencoder-based community detection in complex networks with particle swarm optimization and continuation algorithms,” Journal of Intelligent & Fuzzy Systems, vol. 40, no. 3, pp. 4517–4533, Mar. 2021, doi: 10.3233/jifs-201342.
[22]. M. Al-Andoli, W. P. Cheah, and S. C. Tan, “Deep learning-based community detection in complex networks with network partitioning and reduction of trainable parameters,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 2, pp. 2527–2545, Jul. 2020, doi: 10.1007/s12652-020-02389-x.
[23]. M. U. D. Khanday, M. A. Wani, S. T. Rabani, and Q. R. Khan, “Hybrid Approach for Detecting Propagandistic Community and Core Node on Social Networks,” Sustainability, vol. 15, no. 2, p. 1249, Jan. 2023, doi: 10.3390/su15021249.
[24]. Aldabobi, A. Sharieh, and R. Jabri, “An improved Louvain algorithm based on Node importance for Community detection,” J Theor Appl Inf Technol, vol. 100, no. 23, pp. 1–14, 2022.
[25]. R. Shang, W. Zhang, J. Zhang, L. Jiao, Y. Li, and R. Stolkin, “Local Community Detection Algorithm Based on Alternating Strategy of Strong Fusion and Weak Fusion,” IEEE Trans Cybern, vol. 53, no. 2, pp. 818–831, Feb. 2023, doi: 10.1109/TCYB.2022.3159584.