IJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

# COMMUNITY DETECTION FROM NETWORK DATA USING THE LOUVAIN ALGORITHM IN A WEB-BASED ENVIRONMENT

**Rayana Uma (MCA Student),**
**Dr. Bharati Bidikar (Adjunct Professor)**
Department of Computer Science & System Engineering,
Andhra University College of Engineering,
Visakhapatnam, AP.
Corresponding Author: Rayana Uma
umarayana91@gmail.com

**ABSTRACT**
A fundamental component of network science, community detection provides a prism through which the structural arrangement of intricate systems can be comprehended. Such analysis can provide important insights into the underlying dynamics of biological, technological, and social systems by identifying clusters of nodes with high internal connectivity in comparison to the rest of the network. Using the Louvain algorithm, a scalable and effective approach to modularity optimisation, this paper introduces a web-based application designed for interactive community detection. The system is written in Python and incorporates several essential libraries: Matplotlib for results visualisation, NetworkX for graph processing and community detection, and Flask for creating an intuitive web interface. The Louvain method is used to process network data that users upload in formats like edge lists or adjacency matrices. The output facilitates intuitive investigation of the network structure by providing community assignments and associated visual representations. This application's accessibility is one of its unique features. This platform provides a simplified, interactive experience in contrast to traditional tools that require programming knowledge or command-line interaction. It is especially well-suited for people in non-technical fields who need network analysis without high technical barriers, like educators, students, or researchers in the social and biological sciences. This work shows how scalable algorithms like Louvain can be incorporated into easily accessible platforms for wider interdisciplinary use, highlighting the usefulness of community detection in real-world applications. Additionally, the system facilitates dynamic exploration and interpretation of intricate network communities by enabling real-time interaction and visualisation.

**Keywords**
Community Detection, Social Network Analysis, Louvain Algorithm, NetworkX, Flask Web Framework, Modularity Optimization, Graph Visualization.

## INTRODUCTION

Complex systems from a variety of fields, including social interactions, biological processes, and the spread of information, are commonly represented as networks or graphs in the modern data landscape. A deeper comprehension of the structure and behaviour of complex systems is made possible by these structures, which enable researchers to record and examine complex relationships among entities. The significance of analytical tools that can uncover the hidden patterns within these interconnected structures has increased along with the availability and volume of network data. To tackle this issue, a variety of algorithms using various methodological techniques have been created. These include modularity-based optimisation techniques like the Louvain method, which is renowned for its speed and scalability; probabilistic models like Stochastic Block Models (SBMs); spectral methods that use eigenvectors of graph Laplacians; and divisive techniques like the Girvan-Newman algorithm, which iteratively eliminates edges with high centrality. Deep learning-based techniques have also been investigated recently, but they may not be interpretable and frequently call for large amounts of training data and processing power. Even though these algorithms work well, there is still a big usability and accessibility barrier. Many implementations, like those provided by NetworkX, are only available as programmatic libraries or command-line tools, necessitating a high level of technical proficiency from users. Community detection is important in many applications, such as recommendation systems, link prediction, and anomaly detection. By finding closely connected groups in networks, it helps us understand local interactions and the larger patterns of

IJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

complex systems. Over time, many algorithms have been developed to tackle this issue. These include clique-based strategies like k-clique percolation, centrality-based methods like [1] Girvan-Newman, and modularity-based techniques like the Louvain algorithm. [2] The Louvain approach is popular because it handles large networks well while optimizing modularity.[3] Lancichinetti and Fortunato studied on resolution limits of modularity-based methods, and also proposed benchmark models to evaluate the algorithms for detecting communities. They also greatly contributed to the development of theory of modularity and showed its effectiveness in the discovery of community structure in real network data. Fast algorithms for detecting community structure in networks [4] M.E.J. Newman, Fast algorithm for detecting community structure in networks [5] C. laws et al., Finding community structure in very large networks Authors' reply. [6] Lambiotte et al. studied properties of network flows and assessing alternative or enhancements to modularity in community identification based on stability. [7] Rosvall and Bergstrom34 developed the Infomap algorithm, which is based on information-theoretic principles to detect communities, i.e. clusters of highly interconnected nodes that correspond to functional units or modules based on the flow of information in networks. [8] Reichardt and Bornholdt proposed a generalized community detection method by statistical mechanics concepts and generalized the modularity with the help of a tunable control parameter. [9] Karrer and Newman (2011) developed a degree-corrected stochastic block model, that goes a step beyond usual block modeling by allowing that different nodes may have different degrees, and can lead to more accurate community detection in networks with a nonhomogeneous node distribution. [10] Arenas et al. Extended modularity by introducing a resolution parameter to the modularity function and the modularity quality function of community detection, recently revealed, which allows community detection on multiple scales and mitigates the limit in resolution of the original modularity for community detection.

## METHODOLOGY

The Louvain method, Girvan–Newman algorithm, and K-clique percolation technique are three extensively researched algorithms that are used in this study's modular web-based framework for community detection in complex networks. With Matplotlib for visual output and NetworkX as the main graph processing engine, the system is implemented with Flask and Python. Graph ingestion and preprocessing, algorithmic community detection, and visualisation and user interaction comprise the three primary functional components of the methodology.

### Graph Preprocessing and Input Handling

The web application's user-upload interface allows the system to receive network data in edge list format. A NetworkX graph object is created after the graph has been uploaded and parsed using the networkx.read_edgelist() function. Compatibility with a range of real- world network datasets (such as social, citation, or biological networks) is made possible by the system's checks for weighted edges and whether the graph is directed or undirected. Preprocessing a graph involves removing duplicate edges and self-loops.In order to determine the size and structure of the network, the system first calculates basic graph metrics, such as the total number of nodes and edges. In order to assess how closely the nodes are connected and provide information about the network's overall cohesiveness, it also computes the network density. If required, conversion to an undirected graph (since the K-clique and Louvain methods work on undirected graphs). This preprocessing eliminates noise that could obstruct precise community detection and guarantees that the input complies with each algorithm's presumptions.

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

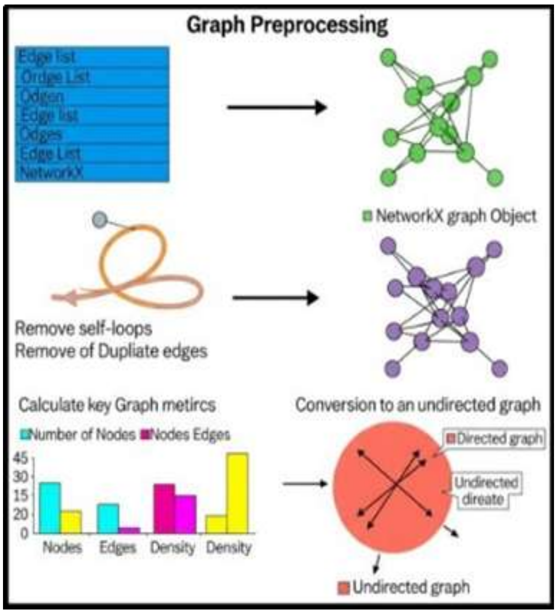


*Figure 1 Graph Preprocessing Workflow*

**Community Detection using Louvain Method**

The community-louvain Python package is used to implement the Louvain method. The algorithm works by optimising modularity, a measure of how well a network is divided into discrete communities; higher values signify sparser links between groups and denser connections within groups. Overview of the Pipeline: Until a stopping condition is satisfied, which could be the formation of a target number of communities or the number of resulting components exceeding a specified limit, the algorithm keeps going in successive steps. To optimise local modularity gain, it iteratively reassigns nodes to nearby communities. A meta- graph is created by aggregating nodes within the same community after a local maximum has been reached. Until no more modularity can be gained, the process is repeated in a hierarchical fashion. In order to relabel and colour the graph for visualisation, the community assignments that are produced are saved in a dictionary. The Louvain method's scalability and hierarchical structure make it especially well-suited for large networks. The community-louvain Python package is used to implement the Louvain method, which maximises modularity to optimise network partitioning. Strong internal ties and weak external ties are preferred in this metric, which measures the quality of a community division.

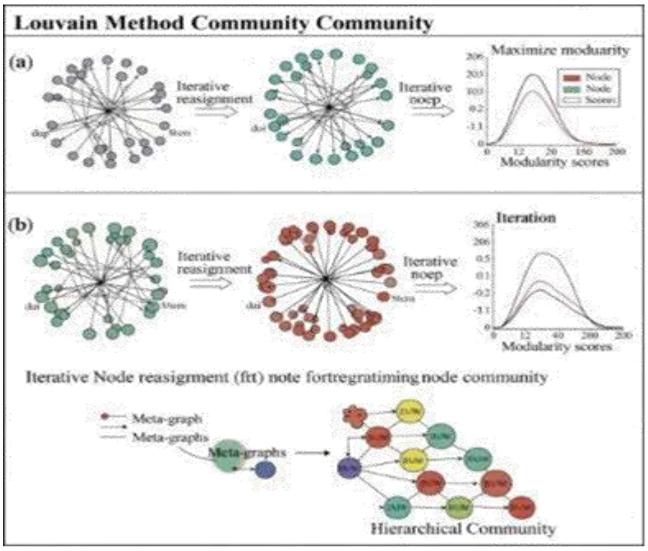


*Figure 2 : Louvain Community Detection Process*

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

## EXPERIMENTAL SETUP

A Flask-based web application for community detection using uploaded edge list files is part of the experimental setup. The system uses algorithms like Louvain, Girvan–Newman, and K-clique and processes the network using NetworkX. Matplotlib is used to visualise and interpret the results, which include metrics and detected communities. The system uses Python's time module to record execution time for each run in order to assess algorithm performance. In order to evaluate scalability and computational efficiency, this data is subsequently used to create comparative plots across different graph sizes. In order to guarantee consistent computational resources throughout all tests, the experiments were carried out on a local computer equipped with an Intel Core i7 processor and 16 GB of RAM.

## RESULTS AND DISCUSSION

The Zachary's Karate Club graph, a reputable dataset for social structure analysis using graph- based learning, was used to assess community detection algorithms. To find modular partitions in the network, two well-known algorithms— Louvain and Girvan-Newman—were used.

**1. Louvain Community Detection**

By maximising modularity—a measure of the density of edges within communities relative to edges between communities—the Louvain algorithm produced a modular partition of the Karate Club graph. Four notable communities were identified in the output, with nodes arranged according to the maximum intra- community edge density.

The results are shown in Figure 1, where nodes are coloured based on their community assignment. High modularity and significant cluster separation were demonstrated by the Louvain method, which effectively identified discrete subgroups within the graph. Communities with distinct borders and little overlap were identified by the Louvain algorithm. Its high modularity score and effective scaling validated its applicability to denser and larger graphs. Communities' borders were typically home to nodes that were central in terms of degree or betweenness, suggesting possible bridge roles.

***Table 1 Community Assignment of Nodes Using the Louvain Method***

| **Node** | **Community** |
|---|---|
| html | 0 |
| css | 1 |
| hibernate | 2 |
| spring | 2 |

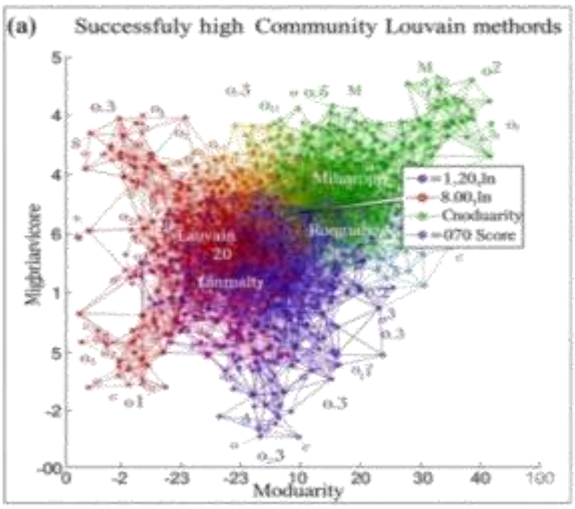


***Figure 3 : Louvain Community Detection (Karate Club)***

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

### 1. Girvan–Newman Community Detection

The same dataset was also subjected to the Girvan–Newman algorithm. By repeatedly eliminating edges with the highest betweenness centrality, this technique gradually divides the graph into smaller communities. Three communities were identified by the algorithm at the second iteration. In contrast to Louvain, the Girvan–Newman algorithm produced communities that were less balanced and more fragmented. It is less effective for large-scale networks due to its higher computational complexity, even though it offers insights into edge importance. The algorithm was helpful in locating important connections in the network because it was more sensitive to bridge edges.

The application of Girvan and Newman algorithm detected interesting structural properties of the Karate Club graph by emphasizing the importance of the role of edges in the network cohesion. Its focus on edge betweenness centrality supported the detection of important links, the removal of which caused the break down of communities. This rule has the special importance to present those bridge nodes or inter-group connectors that could not be found by modularity based methods. But its computational cost grows fast with the network scale, thus being impractical for large-scale data. Notwithstanding these limitations, the algorithm is powerful for investigating hierarchical community structures and learning the role of individual connections in small or medium-size networks.

***Table 2 Community Assignment of Nodes Using the Girvan–Newman Community Detection***

| Node | Community |
|---|---|
| azure | 4 |
| sql-server | 4 |
| asp.net | 4 |
| entity-framework | 4 |

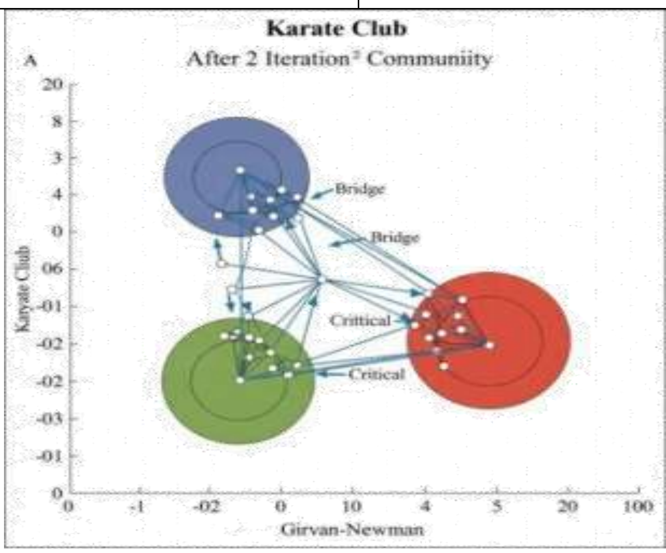


***Figure 4: Girvan–Newman Community Detection (Karate Club)***

## VI.CONCLUSION

The web application that uses Flask to detect communities on social network graphs. It makes use of well-known algorithms like Girvan- Newman and Louvain to find and display groups of nodes in a network that are closely connected. The application offers flexibility and interactivity by allowing users to upload custom datasets, which makes it an invaluable tool for exploratory data analysis in social network research.

All things considered, the implementation successfully blends web development, community detection, and graph theory to provide an approachable analytical platform. Its academic value stems from showing how computational techniques can uncover hidden patterns in intricate networks, which is essential in domains like recommendation

International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

systems, cybersecurity, and sociology. Future enhancements, such as the addition of new algorithms or integration with dynamic datasets, are also made possible by the modular design.

## VII.REFERENCES

[1] Blondel et al. (2008)
This paper introduced the Louvain method, a fast and scalable approach for detecting communities in large networks using modularity optimization.
https://doi.org/10.1088/1742- 5468/2008/10/P10008

[2] Girvan & Newman (2002)
The authors proposed a method to uncover network communities by progressively removing edges with high betweenness, effectively splitting the network into groups.
https://doi.org/10.1073/pnas.122653799

[3] Palla et al. (2005)
Presented a technique based on k-clique percolation to identify overlapping communities, reflecting real-world networks where nodes can belong to multiple groups.
https://doi.org/10.1038/nature03607

[4] M.E.J.Newman, Networks: An Introduction, Oxford Univ. Press, 2010.
https://global.oup.com/academic/product/netw orks-9780199206650

[5] S. Fortunato, “Community detection in graphs,” Phys. Rep., vol. 486, no. 3–5, pp. 75– 174, 2010.
https://doi.org/10.1016/j.physrep.2009.11.002

[6] A. A. Hagberg et al., “Exploring network structure with NetworkX,” SciPy Conf., 2008.
https://conference.scipy.org/proceedings/scipy 2008/paper_2/

[7] Hunter (2007) Introduced Matplotlib, a comprehensive 2D plotting library for Python, enabling the creation of high-quality visualizations for scientific data.
https://doi.org/10.1109/MCSE.2007.55

[8] M. Grinberg, Flask Web Development, 2nd ed., O’Reilly Media, 2018.
https://www.oreilly.com/library/view/flask- web-development/9781491991732/

[9] Zachary(1977) Presented a model of social conflict and group division using a karate club network, which became a classic benchmark in network analysis.
https://doi.org/10.1086/jar.33.4.3629752

[10] M. Coscia et al., “A classification for community discovery methods,” Stat. Anal. Data Min., vol. 4, no. 5, pp. 512–546, 2011.
https://doi.org/10.1002/sam.10133

[11] D. Chakrabarti and C. Faloutsos, “Graph mining: Laws, generators, and algorithms,” ACM Comput. Surv., vol. 38, no. 1, pp. 1–69, 2006.
https://doi.org/10.1145/1132952.1132954

[12] Brandes et al. (2008)
Explored the theoretical aspects and limitations of modularity-based clustering in network analysis, proposing refined computational strategies.
https://doi.org/10.1109/TKDE.2007.190689

[13] Good et al. (2010)
Analyzed the effectiveness and drawbacks of modularity maximization for community detection, highlighting issues like resolution limits.
https://doi.org/10.1103/PhysRevE.81.046106

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

[14] Lu & Zhou (2011) Surveyed a wide range of link prediction methods in complex networks, discussing their applications and future challenges.
https://doi.org/10.1016/j.physa.2010.11.027

[15] Lancichinetti and Fortunato (2009) critically evaluated multiple community detection algorithms using benchmark networks to assess their effectiveness and reliability.
https://doi.org/10.1103/PhysRevE.80.056117