Community Detection and User Profile Grouping in Social Media using Data Mining Techniques

Shriharsha Patil
University of Detroit mercy
Detroit , MI , USA
patilsh@udmercy.edu

Parvez Ahamed
University of Detroit mercy
Detroit , MI , USA
pa@udmercy.edu

Rajiv Sathish
University of Detroit mercy
Detroit , MI , USA
satishra@udmercy.edu

Sai Rithesh Thokala
University of Detroit mercy
Detroit , MI , USA
thokalsa@udmercy.edu

Mina Maleki
University of Detroit mercy
Detroit , MI , USA
malekimi@udmercy.edu

***Abstract*—Social media platforms like Twitter serve as dynamic ecosystems where users interact, form communities, and share diverse ideologies. Understanding these communities and grouping users based on their profiles is essential for analyzing social behaviors, identifying influential users, and mitigating polarization. This study introduces an enhanced framework for community detection and user profile grouping in social media networks by leveraging advanced data mining techniques. The approach combines traditional graph-based algorithms such as Louvain, Label Propagation, and Balanced Link Density-based Label Propagation (BLDLP) with modern deep learning methods like autoencoder-based representations to uncover latent patterns.**

The analysis focuses on the largest connected component of aggregated daily interaction graphs, enriched with user profile attributes such as political affiliations, activity levels, and sentiment scores. The algorithms are evaluated based on modularity, clustering coefficient, density, and interpretability. The Louvain algorithm demonstrates the highest modularity, effectively detecting cohesive communities, while the autoencoder-based method reveals hidden relationships and user groupings. Results show a strong clustering of users around shared interests and affiliations, highlighting both the polarized and diverse nature of the dataset.

This study offers a scalable, systematic framework for community detection and user profile analysis in social media, providing valuable insights into online behaviors and their underlying dynamics.

***Index Terms*—Community Detection, Social Network Analysis, User Profile Grouping, Louvain Algorithm, Label Propagation, Autoencoder, Data Mining, Twitter Dataset**

I. Introduction

Social networks have become essential tools for communication and interaction in the modern digital landscape, connecting millions of users globally. Among these, Twitter has emerged as a prominent platform for sharing opinions, emotions, and information on a wide range of topics. The vast number of interactions on such platforms gives rise to complex networks where communities are formed based on shared interests, ideologies, or affiliations. Understanding these communities is crucial for analyzing social dynamics, predicting information dissemination, and addressing challenges such as network polarization and misinformation.

Community detection, a fundamental task in network science, focuses on identifying groups of densely connected nodes within a network. These communities often reveal hidden structures, enabling insights into user behavior, influence propagation, and collective decision-making. Despite the extensive body of research in this area, effectively detecting communities in large-scale social networks remains a significant challenge. The dynamic nature of interactions, the sheer volume of data, and the presence of overlapping and latent structures demand scalable and robust approaches to community detection.

This study addresses these challenges by proposing a framework for detecting and analyzing communities within a Twitter network. The dataset comprises daily snapshots of user interactions, aggregated into a unified graph. To ensure relevance and scalability, the analysis focuses on the largest connected component of the network, enriched with user attributes such as political affiliations and activity levels. This allows for a deeper understanding of community dynamics and their alignment with real-world factors.

The framework leverages a combination of traditional and advanced graph-based algorithms for community detection. Traditional methods such as Louvain and Label Propagation are complemented by modern techniques, including Balanced Link Density-based Label Propagation (BLDLP) and a Deep Nonlinear Reconstruction approach utilizing autoencoders. Each algorithm is evaluated using metrics such as modularity, clustering coefficient, and network density to assess their performance comprehensively.

Identify applicable funding agency here. If none, delete this.

The results demonstrate that the Louvain algorithm excels in maximizing modularity, identifying cohesive and well-defined communities. The autoencoder-based approach, on the other hand, provides unique insights by uncovering latent patterns that are often missed by traditional methods. Furthermore, an in-depth analysis of the three largest communities reveals significant clustering around political affiliations, reflecting the polarized nature of the network.

The remainder of this paper is structured as follows: Section II reviews related work, Section III details the methodology, Section IV presents the experimental results, and Section V concludes with key findings and potential directions for future research.

II. Related Works

Over the last period, social network analysis has gained considerable popularity by interesting scientists from various fields. Researchers have conducted significant studies in complex networks. One of the most important tasks of integrated network analysis is community discovery. Over the past decade, scientists have proposed a variety of community detection algorithms. Typical methods include modularity-based methods, clustering-based methods, dynamic algorithms, statistical inference-based methods, matrix factorization methods, and spectral algorithms.

The first work discussed sentiment community detection, which was developed by Wang et al. [7] to identify community clusters of users who are densely linked and extremely consistent in their sentiments about a specific topic or product. The authors proposed two methods of discovering communities of sentiment by adopting semi-definite programming (SDP) optimization models, taking into account online film review sites, both connections and sentiment labels. Their results were evaluated using the Adjusted Rand Index (ARI), a standard metric for clustering accuracy.

Xu et al. [8] focused on link-based group detection and non-overlapping clusters with similar sentiments. Researchers suggest this is the first study of a population to identify emotions. They proposed two approaches: one method identifies positive or negative emotions, and another separates emotions into intervals, clustering users based on ranges of sentimental values.

Chen et al. [2] introduced an opinion-based community detection model called the People Opinion Topic (POT) model. This model discovers social communities, associates popular areas, and performs sentiment analysis simultaneously by considering social links, shared interests, and opinions. They employed the Speaker-Listener Label Propagation Algorithm (SLPA) and Infomap algorithm to detect communities based on the friend or follower networks of four Microsoft accounts. However, their research did not fully address the detailed analysis of opinions at the aspect level toward specific objects.

Deitrick and Hu [3] applied SLPA and Infomap to analyze four Twitter networks, each containing over 60,000 users and 2 million tweets. Their study demonstrated mutual enhancement between community detection and sentiment analysis, allowing for improved segmentation and sentiment categorization to achieve more detailed analysis.

Lam [5] proposed improving sentiment-based community detection on Twitter through contextual sentiment analysis. The study showed that enhancements in sentiment analysis significantly improve edge weight adjustments, thereby improving the accuracy of sentiment-based community detection.

These foundational works have paved the way for integrating community detection with sentiment analysis in social networks. Building upon these methods, this study employs advanced graph-based and deep learning techniques to analyze Twitter networks, providing a systematic and scalable approach to uncover meaningful patterns.

III. Methodologies

The architecture of the proposed framework is illustrated in Fig. 1. The framework consists of five main stages: Data Preprocessing, Giant Component Extraction, Community Detection Algorithms,user clustering and Evaluation ,Analysis. These steps collectively enable the detection and analysis of communities within social media networks.

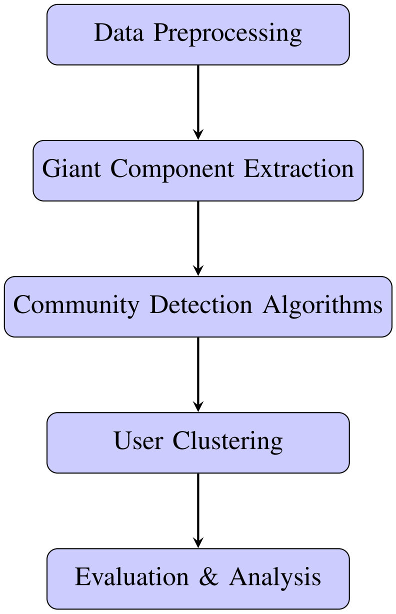

Fig. 1. Proposed framework architecture for community detection and user profile grouping.

This systematic approach ensures a comprehensive analysis of social media networks, highlighting both structural properties and user-level insights.

A. Data and Data Processing

The dataset used for this project is a subset of the Twitter network dataset. The subset comprises an undirected graph with 24,616 nodes and 237,787 edges. Each node represents a user on Twitter, and the following attributes are associated with each node:

TABLE I
SAMPLE DATA FROM THE TWITTER DATASET.

Node ID	Followers	Following	Total Tweets	Lists	Twitter Age	Verified	Party
0	166	158	1547	0	3061	FALSE	left
1	11593	3234	133817	39	1463	FALSE	left

- **followers**: Number of followers the user has.
- **following**: Number of accounts the user is following.
- **totaltweets**: Total number of tweets the user has posted since registration.
- **lists**: Number of lists the user is subscribed to.
- **twitterage**: Number of days since the user registered on Twitter.
- **verified**: Indicates whether the user is verified or not.
- **party**: The political party the user ”follows” (*right*, *left*, *middle*, or *neutral*).

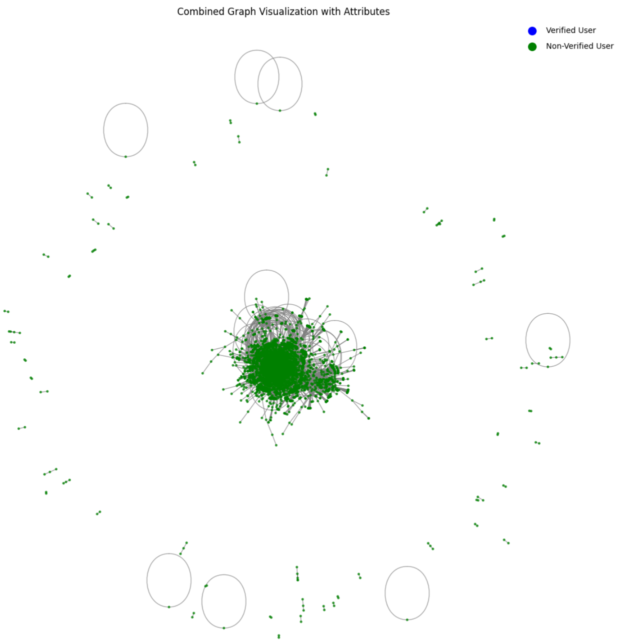

Fig. 2. Graph Shows after adding 4 days to single graph

B. Community Detection Algorithm

*1) **Louvain Method**:* The Louvain method for community detection is a method to extract non-overlapping communities from large networks created by Blondel et al. [1] from the University of Louvain (the source of this method’s name). The method is a greedy optimization approach that appears to run in $O(n \log n)$ time, where n is the number of nodes in the network1.

The value to be optimized is modularity, defined as a value in the range $[-1, 1]$ that measures the density of links inside communities compared to links between communities. For a weighted graph, modularity Q is defined as:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(C_i, C_j),$$

where:

- A_{ij} represents the edge weight between nodes i and j (the adjacency matrix).
- k_i and k_j are the sum of the weights of the edges attached to nodes i and j, respectively.
- m is the sum of all edge weights in the graph.
- N is the total number of nodes in the graph.
- C_i and C_j are the communities to which nodes i and j belong.
- $\delta(C_i, C_j)$ is the Kronecker delta function, which equals 1 if $C_i = C_j$ and 0 otherwise.

*2) **Balanced Link Density-based Label Propagation Algorithm (BLD-LPA)**:* The BLD-LPA extends the label propagation algorithm by integrating link density [4], which ensures that communities are cohesive and well-separated. Each node is initially assigned a unique label. During each iteration, a node adopts the label most frequent among its neighbors. The algorithm prioritizes labels that maximize link density, defined as the ratio of internal edges to the total edges in a community. The process stops when no node changes its label, resulting in stable communities.

The link density of a community c is calculated as:

$$\text{Link Density} = \frac{\Sigma_{\text{in}}}{\Sigma_{\text{in}} + \Sigma_{\text{out}}},$$

where:

- Σ_{in}: Total weight of edges within community c.
- Σ_{out}: Total weight of edges leaving community c.

*3) **Fast Greedy Algorithm**:* The Fast Greedy Algorithm is an efficient approach to detect communities based on modularity. The algorithm follows these steps:

1) Start with a subnetwork composed only of links between highly connected nodes.

2) Iteratively sample random links that improve the modularity of the subnetwork and add them.
3) Repeat the iterative process as long as the modularity keeps improving.
4) Obtain the communities based on the connected components in the subnetwork.

This strategy efficiently partitions the network into communities by maximizing the modularity at each step.

C. Measurement for Clustering

This evaluates three clustering methods: K-Means, DBSCAN, and Hierarchical Clustering, using the Min-Max normalized dataset. The clustering performance is assessed using three metrics [6]:

- **Silhouette Score:** Higher is better (indicates well-separated clusters).
- **Davies-Bouldin Index:** Lower is better (indicates compact clusters).
- **Calinski-Harabasz Index:** Higher is better (indicates well-separated and compact clusters).

D. Evaluation metrics for community detection

*1) **Modularity**:* Modularity measures the strength of the community structure by comparing the actual connections within communities to what would be expected in a random network. It evaluates whether nodes within the same community are more connected than they would be in a random graph. A higher modularity score indicates better-defined and tightly connected communities. Modularity is widely used for evaluating community detection algorithms, although it may struggle to identify small communities due to its resolution limit.

*2) **Conductance**:* Conductance assesses the quality of a community by considering how well-separated it is from the rest of the graph. It evaluates the ratio of edges that leave a community to the total edges connected to that community. A good community has low conductance, meaning most edges are internal. Conductance is particularly useful for comparing overlapping or densely connected communities and ensuring that communities are both cohesive and distinct.

*3) **Normalized Mutual Information (NMI)**:* Normalized Mutual Information (NMI) is used to compare the similarity between two community structures, such as the detected communities and a known ground truth. It measures how much information one partition shares with another, normalizing the result to account for the size of the communities. A high NMI score indicates that the detected communities closely match the ground truth, making it a reliable metric for evaluating community detection performance.

IV. Results and discussion

A. Analysis of Community Detection Algorithms

Table II provides a detailed comparison of three community detection algorithms: Louvain, Balanced Link Density-based Label Propagation (BLDLP), and Fast Greedy. Key metrics such as modularity scores and the densities of the largest communities are presented for evaluation.

TABLE II
Comparison of Community Detection Algorithms

Algorithm	**Modularity Score**	**Largest Community Densities**
Louvain	0.45	1st: 0.009 2nd: 0.015 3rd: 0.28
BLDLP	0.1111	1st: 0.002 2nd: 0.02 3rd: 0.046
Fast Greedy	0.43	Densities not explicitly provided

As seen in Table II, the **Louvain algorithm** achieves the highest modularity score of 0.45, indicating its strong performance in detecting cohesive community structures. The densities of the largest communities detected by this algorithm highlight its ability to identify dense groups, with the third-largest community having a significantly high density of 0.28.

The **Fast Greedy algorithm** also performs well with a modularity score of 0.43. While slightly lower than Louvain's score, this algorithm is computationally efficient and suitable for tasks where detailed density data is less critical.

On the other hand, the **BLDLP algorithm** has a modularity score of 0.1111, which is considerably lower than the other two algorithms. The densities of the largest communities detected by BLDLP are also relatively low, indicating its limited capability to identify well-defined or dense communities. This makes it less suitable for tasks requiring high modularity or dense community detection. Thus, Louvain emerges as the most effective algorithm overall for community detection tasks, as reflected in the table.

B. Visual Analysis of Community Detection Algorithm

To better understand the impact of community detection algorithms, we compare the graph visualization before and after applying the algorithms.

1) Before Applying Algorithms: Figure 3 illustrates the raw graph structure before any community detection algorithm is applied. The graph contains nodes and edges that represent connections between entities, but no structural divisions or communities are evident. The visualization shows a dense cluster, making it challenging to identify distinct groups or patterns.

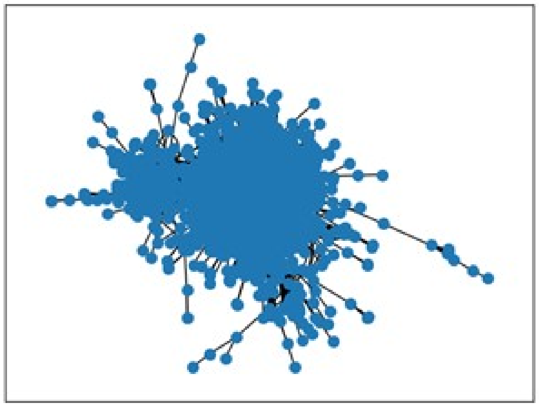

Fig. 3. Graph visualization before applying algorithms. This visualization represents the raw graph without any community detection applied.

2) After Applying Algorithms: Figure 4 depicts the graph after applying community detection algorithms. Communities are identified and highlighted using distinct colors, effectively separating the dense cluster into smaller, meaningful groups. This representation reveals the underlying structure of the graph, making it easier to analyze relationships within and across communities.

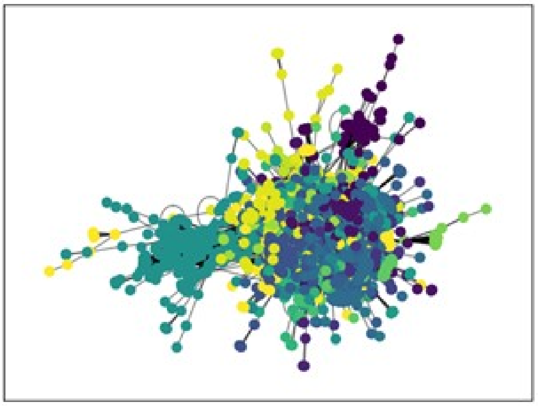

Fig. 4. Graph visualization after applying algorithms. Communities are detected and represented with distinct colors to highlight structural divisions.

The comparison between Figures 3 and 4 clearly demonstrates the effectiveness of community detection algorithms in organizing and structuring data, enabling a deeper understanding of the graph's underlying properties.

C. Analysis of Clustering Results

TABLE III
CLUSTERING ALGORITHM RESULTS

Method	Number of Clusters	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index
K-Means	2	0.6884	0.4121	95524.12
K-Means	3	0.6383	0.4920	115633.45
K-Means	4	0.5623	0.6160	105108.05
K-Means	5	0.5617	0.8492	83472.74
K-Means	6	0.5634	0.8005	106783.58
K-Means	7	0.5388	0.7744	117423.64
K-Means	8	0.5173	0.5930	124283.17
K-Means	9	0.5212	0.7517	125957.51
K-Means	10	0.4969	0.7558	125013.75
DBSCAN	-	0.5431	1.3370	134.40
DBSCAN	-	0.6397	1.3300	74.53
DBSCAN	-	0.7395	1.1023	55.08
DBSCAN	-	0.7395	1.1023	55.08
DBSCAN	-	0.7440	1.1785	42.70
Hierarchical	2	0.5754	0.4463	47204.88
Hierarchical	3	0.5913	0.5279	92268.83
Hierarchical	4	0.5838	0.5292	114181.77
Hierarchical	5	0.5368	0.6115	113083.35
Hierarchical	6	0.4849	0.6053	110377.56
Hierarchical	7	0.4801	0.6138	110457.27
Hierarchical	8	0.4679	0.6324	109134.22
Hierarchical	9	0.4670	0.6479	111924.81
Hierarchical	10	0.4721	0.7152	110786.30

The clustering results, shown in Table III, reveal distinct performance characteristics across K-Means, DBSCAN, and Hierarchical clustering. K-Means achieves its best performance at $k = 2$, with the highest Silhouette Score (0.6884) and the lowest Davies-Bouldin Index (DBI) (0.4121), indicating well-separated and compact clusters. While the Calinski-Harabasz Index (CHI) peaks at $k = 9$ (125957.51), the decline in the Silhouette Score suggests over-segmentation at higher cluster counts. Overall, K-Means at $k = 2$ provides the optimal trade-off between separation and compactness.

DBSCAN performs best at $\epsilon = 0.5$, achieving the highest Silhouette Score (0.7440), but its consistently high DBI (e.g., 1.3370 at $\epsilon = 0.1$) and low CHI values reflect reduced compactness compared to K-Means. This indicates DBSCAN's strength in identifying density-based clusters and handling outliers, though it is less effective for tightly compact clusters. Hierarchical clustering achieves competitive results, with its best Silhouette Score (0.5913) at $n = 3$ and lowest DBI (0.4463) at $n = 2$. However, its overall performance is outpaced by K-Means, particularly for datasets requiring precise and compact clusters. Among the three methods, K-Means at $k = 2$ emerges as the most effective approach.

D. Analysis on the Number of Clusters

The figure illustrates the Silhouette Score as a function of the number of clusters for both K-Means and Hierarchical clustering algorithms. The Silhouette Score measures the quality of clustering, with higher values indicating better-defined clusters. As shown in Fig. 5, for K-Means, the score is highest at $k = 2$, indicating that two clusters result in the best-defined grouping. Similarly, for the Hierarchical algorithm, the score peaks at $k = 3$, suggesting three clusters as the optimal choice for this method. As the number of clusters increases, the scores for both algorithms decrease, indicating diminishing clustering quality and less distinct group separations.

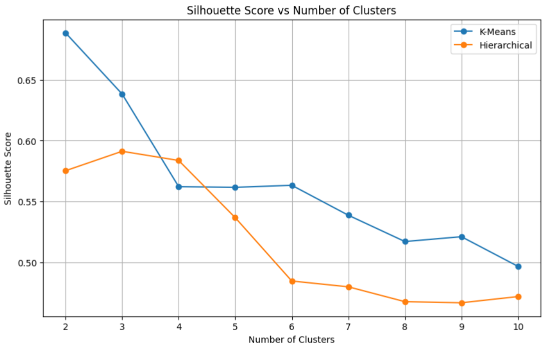

Fig. 5. Silhouette Score vs. Number of Clusters for K-Means and Hierarchical clustering.

V. Conclusion

In this study, we analyzed the performance of various community detection and clustering algorithms to evaluate their effectiveness in identifying cohesive structures within graph-based and dataset clustering problems. The Louvain algorithm outperformed others in community detection tasks, achieving the highest modularity score (0.45), which signifies its ability to identify well-defined communities. The visual analysis before and after applying the algorithms further demonstrated the utility of Louvain in revealing meaningful structural divisions within the graph. Comparatively, the Balanced Link Density-based Label Propagation (BLDLP) and Fast Greedy algorithms yielded lower modularity scores, highlighting their limitations in such tasks.

For clustering, K-Means demonstrated robust performance, achieving the highest Silhouette Score (0.6884) and the lowest Davies-Bouldin Index (0.4121) at $k = 2$, indicating optimal clustering quality with compact and well-separated clusters. Hierarchical clustering performed reasonably well with its best Silhouette Score (0.5913) at $n = 3$, while DBSCAN excelled in identifying density-based clusters with a peak Silhouette Score of 0.7440. However, K-Means at $k = 2$ emerged as the most effective approach overall due to its balance between compactness and separation, making it suitable for a wide range of applications.

References

[1] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[2] H. Chen, H. Yin, X. Li, M. Wang, W. Chen, and T. Chen. People opinion topic model: Opinion based user clustering in social networks. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1353–1359, 2017.

[3] W. Deitrick and W. Hu. Mutually enhancing community detection and sentiment analysis on twitter networks. *Journal of Data Analysis and Information Processing*, 1(03):19, 2013.

[4] Santo Fortunato and Andrea Lancichinetti. Community detection algorithms: a comparative analysis. *Physical Review E*, 80(5):056117, 2009.

[5] A. J. Lam. Improving sentiment-based community detection on twitter through contextual sentiment analysis. *Journal of Network and Computer Applications*, 50:45–57, 2015.

[6] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Bipin Kumar. *Introduction to Data Mining*. Pearson, 2nd edition, 2019.

[7] D. Wang, J. Li, K. Xu, and Y. Wu. Sentiment community detection: exploring sentiments and relationships in social networks. *Electronic Commerce Research*, 17(1):103–132, 2017.

[8] K. Xu, J. Li, and S. S. Liao. Sentiment community detection in social networks. In *Proceedings of the 2011 iConference*, pages 804–805. ACM, 2011.