A Graph-Based Hybrid Clustering Approach for Detecting Complex Structures

Nassira Lograda[1/2✉], Makhlouf Benazi[1/3]

[1] *Laboratory of Informatics and its Applications of M'sila, University of M'sila, M'sila, Algeria*
[2] *nassira.logarada@univ-msila.dz*
[3] *makhlouf.benazi@univ-msila.dz*

Abstract

Clustering is essential for identifying patterns in data by grouping similar points. However, many advanced algorithms face challenges when dealing with clusters of varying shapes and sizes. In this paper, we propose a KNN-FN hybrid algorithm combines the strengths of K-Nearest Neighbors (KNN) and the Fast Newman (FN) community detection algorithm to enhance clustering performance. KNN is used to construct a graph that captures local neighborhood structures by connecting each data point to its nearest neighbors, while the FN algorithm applies modularity maximization to detect well-defined clusters within the graph.

This hybrid approach improves clustering, particularly in complex datasets with irregular shapes and varying densities, The KNN-FN hybrid algorithm efficiently detects clusters in large-scale data, making it suitable for real-world applications.

Keywords: *Clustering, Hybrid, K-nearest neighbor, Fast Newman*

1. Introduction

Clustering is a foundational technique in unsupervised learning, widely applied in areas such as data mining, pattern recognition, and machine learning.Unlike supervised learning, where class labels are provided, clustering seeks to uncover the inherent structure of data by grouping similar objects without prior knowledge of their categories. The goal is to divide a dataset into K distinct clusters (where K may be known or estimated) so that data points within the same cluster exhibit high intra-cluster similarity and low inter-cluster similarity. This similarity is typically assessed using distance-based metrics such as Euclidean or cosine distance, or through other measures of association depending on the nature of the data.[1][2]

Several main classes of clustering methods have been developed, including Density-based Clustering (DC), Grid-based Clustering (GC), Model-based Clustering (MC), Hierarchical Clustering (HC), and Partitional Clustering (PC).[3]

Hybrid clustering is an advanced approach that combines the strengths of multiple clustering algorithms or their components to enhance accuracy, stability, and scalability [3][4]. By integrating complementary strategies, hybrid methods achieve greater robustness and adaptability when dealing with complex datasets.Classification of hybrid models can be structured as follows:

1. Component Combination-Based Hybrid Structure: Combines different algorithmic components to create a unified model.
2. Parallel Hybrid Structure: Runs multiple models in parallel, integrating their results for enhanced decision-making.
3. Series Hybrid Structure: Connects models in a sequence where the output of one model feeds into the next.

4. Parallel–Series Hybrid Structure: Combines elements of both parallel and series structures, allowing for simultaneous and sequential processing for improved efficiency.[4]

In this paper, we propose a new hybrid clustering method based on the K-Nearest Neighbors (KNN) and Fast Newman (FN) algorithms. While KNN is generally used for classification and regression, it can be indirectly applied to clustering in certain ways. First, we use KNN to convert the data into a graph. Next, we apply the FN community detection algorithm to this graph.

The rest of this paper is organized as follows: In section 2, we present the basic concept of KNN and FN algorithms. Then, some variants of KNN algorithm in clustering are explained in section 3. Section 4 is devoted to presenting our approach in detail. To show the performance of our method, section 5 gives some results of the application of our method on different datasets well known in the clustering world. Finally, section 6 concludes the paper and gives some perspectives.

2. Backgrounds

2.1 KNN algorithm

The k-nearest neighbor (KNN) algorithm is a supervised learning method commonly used for classification tasks. It classifies an unlabeled instance by identifying its k nearest neighbors in the feature space and assigning the most frequent class label among them [5].

Although traditionally used for supervised learning, the KNN principle can be adapted for unsupervised clustering by constructing a KNN graph, where nodes represent data points and edges connect each point to its k nearest neighbors according to a chosen distance metric (e.g., Euclidean distance). This transformation converts the dataset into a graph structure that captures local neighborhood relationships, making it suitable for community detection or graph-partitioning algorithms [6].

Once the KNN graph is built, graph-based clustering can be performed by applying community detection techniques that group nodes with dense internal connections and sparse external links. Such graph representations are particularly effective for identifying clusters with nonlinear boundaries, varying densities, or noise.[7]

Algorithm 1: KNN Graph construction

Input:
- Dataset D
- Number of neighbors k

Output:
-Gaph
-Build KNN Graph:
For each point p in D:
- Find k nearest neighbors.
- Connect p to its neighbors.

-Return graph

2.2. FN algorithm

The Fast Newman (FN) algorithm is a hierarchical clustering approach designed to detect community structures in networks by optimizing modularity, a measure that reflects the quality of the partitioning. [8] Initially, each node is treated as its own community, and at each step, the algorithm calculates the

modularity gain for merging two communities. The pair that maximizes this gain is merged. This process repeats iteratively until no further mergers increase modularity, resulting in the optimal partitioning.[8] Unlike the Girvan-Newman algorithm, which is computationally expensive, the FN algorithm has a lower time complexity of approximately (O (m(m + n))), making it faster and more efficient for medium-sized networks. However, when applied to large-scale datasets such as social networks or regional call data, where the number of connections can reach millions, even this complexity becomes time-consuming. The FN algorithm uses a greedy strategy to approximate optimal community structures, particularly when dealing with large datasets. Despite its efficiency, it may not always find the absolute best solution but remains a widely used method for network clustering due to its balance between accuracy and computational speed.[10]

Algorithm 2: FN algorithm

Input:
- Graph G with nodes and edges

Output:
- Community structure with maximum modularity

1. **Initialization**
   - Each node is initialized as its own community.

2. Modularity Calculation

2.1 Compute the initial modularity Q for the graph G.
2.2 For each pair of communities, calculate the potential change in modularity if they are merged.

3. Iterative Merging

While there are at least two communities:
3.1 Find the pair of communities (i, j) that leads to the largest increase in modularity Q.
3.2 Merge communities i and j.
3.3 Update the modularity Q based on the new community structure.
3.4 Repeat until no further increase in modularity is possible.

4. **Return** the final community structure with the highest modularity Q.

3. Related work

Recent advancements have aimed at enhancing clustering algorithms to improve their flexibility, efficiency, and effectiveness across diverse data types. Recent developments have focused on improving clustering algorithms to enhance their flexibility, efficiency, and effectiveness across different types of data. Many researchers have been interested in hybridization, including: Shi Bing et al [11] proposed a clustering method that enhances traditional algorithms by integrating k-nearest neighbors (KNN) with a density-based approach. It automatically determines local parameters, detects clusters of arbitrary shapes, and eliminates noise, offering flexibility in clustering datasets with varying densities. The experimental results validate its effectiveness.

Liu et al [12] introduce a graph-theoretical approach to clustering, combining graph construction with Markov Stability for multiscale community detection. The method estimates the number of clusters automatically and shows improved performance compared to traditional clustering methods on synthetic and real datasets.

Li et al [13] propose an adaptive density-based clustering algorithm (ADBSCAN) that identifies high-density core samples using the nearest neighbor graph (NNG) instead of traditional density estimators.

Experimental results on synthetic and real datasets show that ADBSCAN improves performance compared to existing density-based clustering methods.

Research indicates that hybridization has led to effective outcomes in clustering techniques, motivating the development of a new hybrid approach. In the following paragraph, we will introduce our innovative algorithm, which combines KNN and FN

4. Hybrid KNN-FN

In this section, we explain our proposed algorithm, two procedures are proposed in this paper. The algorithm begins by constructing a k-Nearest Neighbors (k-NN) graph, where each data point is represented as a node, and edges connect each node to its k closest neighbors based on a distance metric like Euclidean distance. This step involves calculating pairwise distances between all points, which has a time complexity of $O(n^2)$ for n data points. The result is an adjacency matrix, where entries indicate connections between nodes, ensuring symmetry so that if node A is connected to node B, node B is also connected to node A. Once the k-NN graph is built, community detection is applied to identify groups of strongly connected nodes. This is achieved using greedy modularity optimization, which starts by treating each node as its own community and iteratively merges nodes or communities to maximize modularity, a measure of the density of edges within communities compared to edges between them. The greedy modularity algorithm has a complexity of O(m log n), where m is the number of edges and n is the number of nodes. The process continues until no further improvement in modularity is possible, resulting in well-defined clusters. Finally, the algorithm outputs the community assignments for each node. This approach is widely used in clustering, social network analysis, and biological network analysis to uncover meaningful patterns and groups within data, but its overall complexity is dominated by the $O(n^2)$ distance calculations and the O(m log n) community detection step, making it less scalable for very large datasets.

Algorithm 3: Hybrid KNN-FN

Input:
- Dataset X with n data points.
- Number of neighbors k.

1. **Create KNN Graph**:
   - Initialize and fit the KNN model on the dataset.
   - Generate the k-nearest neighbors graph.
   - Make the graph undirected(if node A is connected to node B, then B is also connected to A) and remove self-loops.
2. **Community Detection**:
   - Apply a community detection algorithm to identify clusters.
   - Assign cluster labels to each data point based on the detected communities.

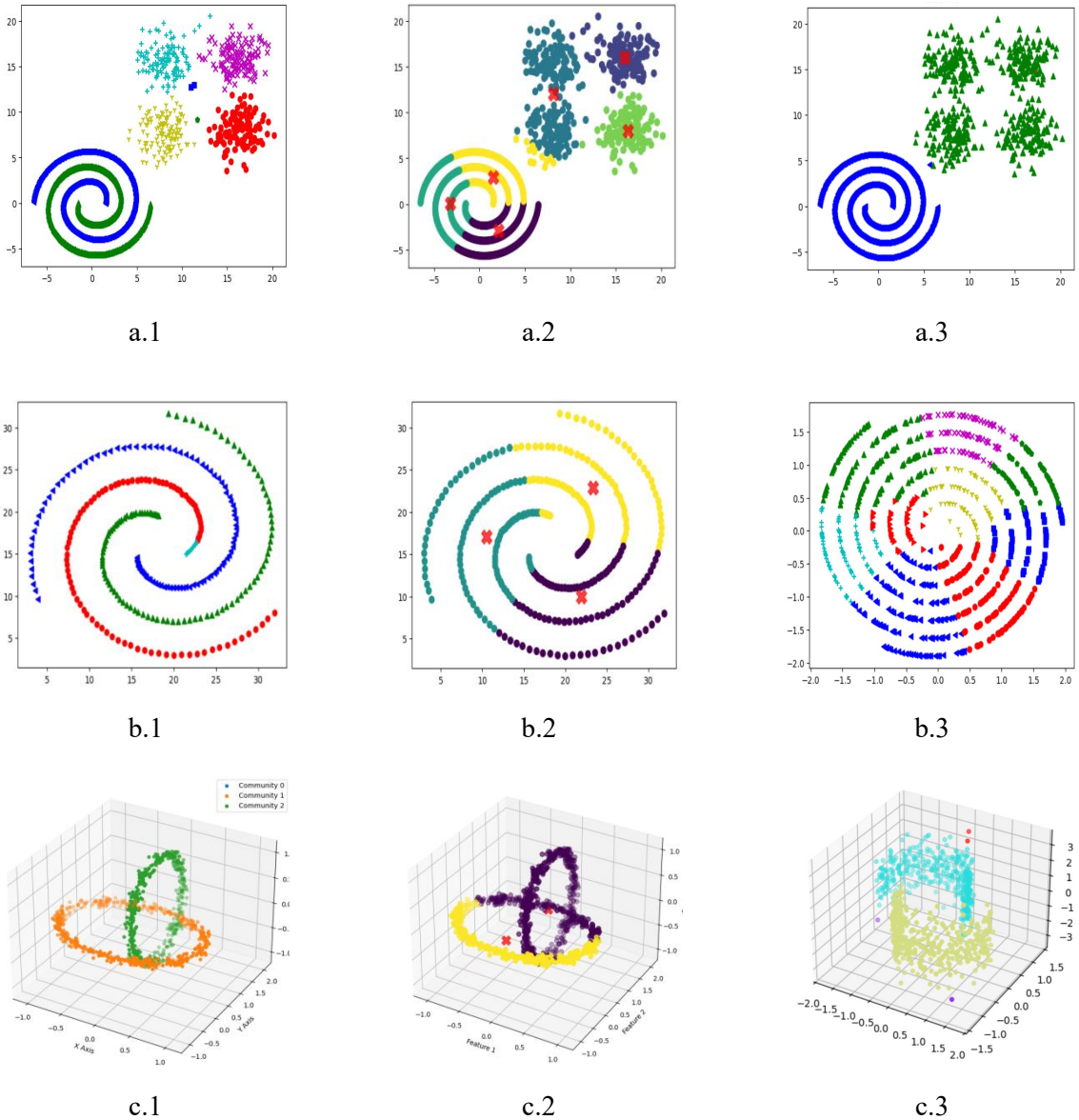
Output:
- Clustered data with labels for each data point.

5. Experimental Evaluation

The objective of this section is to perform comprehensive computational experiments to assess the performance of the proposed hybrid KNN–FN clustering algorithm.
To evaluate its effectiveness, the algorithm was tested on three synthetic datasets and two real-world datasets with varying characteristics in terms of shape, dimensionality, and density.

Table 1. Description of Synthetic and Real-World Datasets

Dataset	Type	Samples	Features	Clusters	Description
SpiralSquare	synthetic	1500	2	6	Spiral + square mixed shapes
3-Spiral	synthetic	312	2	3	Three intertwined spirals
chainlink	synthetic	1000	3	2	Two interlocking rings
wine	Real	178	13	3	Wine chemical composition
yeast	Real	1484	8	10	Protein localization sites



a.1 a.2 a.3

b.1 b.2 b.3

c.1 c.2 c.3

Figure 1.Comparison of Knn-FN and K_means and Infomap

Figure 1 presents a visual comparison between the proposed KNN–FN algorithm and the baseline methods K-Means and Infomap across three synthetic datasets: SpiralSquare, 3-Spiral, and Chainlink. Each row corresponds to a distinct dataset (row a for SpiralSquare, row b for 3-Spiral, and row c for Chainlink), while each column represents a different clustering algorithm: column 1 displays the results of KNN–FN, column 2 shows K-Means, and column 3 illustrates Infomap for comparison.

The results demonstrate that KNN–FN produces more accurate and well-separated clusters, particularly for non-linear and intertwined data structures. In the SpiralSquare and 3-Spiral datasets, KNN–FN successfully preserves the complex geometries that K-Means fails to capture, while Infomap provides only partial separations with overlapping boundaries. For the Chainlink dataset, KNN–FN achieves clear separation of the three interlinked rings in 3D space, outperforming both baseline methods in handling non-convex and spatially complex structures. These observations confirm the robustness, adaptability, and structural sensitivity of the proposed hybrid approach across datasets with diverse geometries and densities.

It is important to note that, for synthetic datasets where the true cluster labels are known and created to show clear geometric shapes, visual inspection is the main way to evaluate results. This type of qualitative analysis focuses on how well the algorithm can reproduce the overall shape, connections, and structure of the data, especially when common numerical measures do not fully describe these patterns. Overall, the results in Figure 1 show that the proposed graph-based KNN–FN method is more flexible and reliable in detecting complex cluster patterns than traditional algorithms.

Table 2. Clustering performance of real datasets

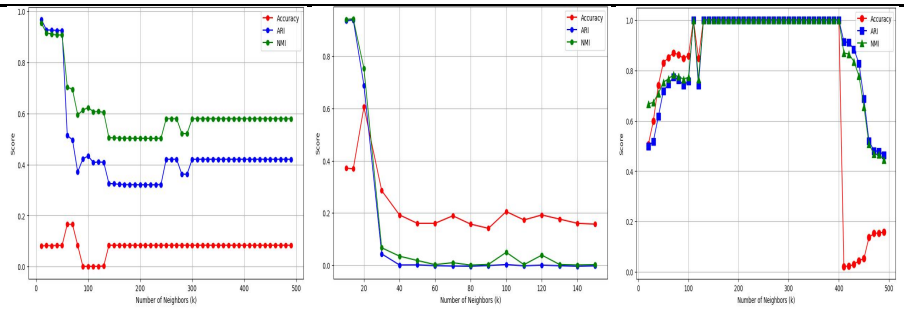
Data / Methods	Wine			Yeast		
	ACC	ARI	NMI	ACC	ARI	NMI
KNN-FN	**0.9157**	**0.8518**	**0.8362**	0.2439	0.1728	**0.2475**
K-Means	0.426966	0.388724	0.501363	**0.277628**	0.126637	0.137559
Infomap	0.292135	0.655794	0.704464	0.026280	**0.134886**	0.206904

Table 2 compares the performance of three clustering methods KNN-FN, K-Means, and Infomap on the Wine and Yeast datasets using three evaluation metrics: Accuracy (ACC), Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI).

For the Wine dataset, KNN-FN clearly outperforms the other methods, reaching high scores across all metrics (ACC = 0.9157, ARI = 0.8518, NMI = 0.8362). This shows that KNN-FN can correctly identify and separate the real clusters in the data. K-Means and Infomap show weaker results, although Infomap performs slightly better in ARI and NMI.

For the Yeast dataset, which is more difficult due to overlapping classes, the performance of all methods decreases. However, KNN-FN still achieves the best overall results, confirming its robustness and ability to handle complex data.

Overall, KNN-FN consistently performs better than traditional clustering algorithms, proving to be more accurate and reliable for both simple and complex datasets.



(1) (2) (3)

Figure 2.Results for determining the optimal number of neighbors on three datasets. (1) SpiralSquare, (2) 3-Spiral, and (3) chainlink.

Figure 2 shows how changing the number of neighbors K affects the clustering results of the proposed KNN-FN algorithm on three synthetic datasets: SpiralSquare (1), 3-Spiral (2), and Chainlink (3).
The performance is measured using three common metrics Accuracy (ACC), Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI) to give a full view of clustering quality and stability.
This experiment shows how sensitive the algorithm is to the choice of K, which controls how the data points are connected in the neighborhood graph.

For the SpiralSquare dataset (1), all three metrics (ACC, ARI, and NMI) change a lot for small values of K, then reach their best values around k≈50 at this point, the algorithm captures both local and global structures effectively.When K becomes larger than about 100, the results start to drop and then stay low. This means that using very large neighborhoods causes nearby clusters to mix together.
So, a medium value of K gives the best balance between compact and well-separated clusters.

In the 3-Spiral dataset (2), the best results for all three metrics appear when K is small, around 10–15.
This shows that the complex spiral shapes are best represented when only close neighbors are used.
As K increases, the performance gradually decreases because the spirals start to overlap and lose their clear boundaries.

For the Chainlink dataset (3), the three metrics increase quickly with K, reaching near-perfect clustering between k≈50 and 400.This suggests that stronger connections help the algorithm capture the global structure of the two linked rings. However, when K becomes larger than about 400, the metrics drop sharply again, meaning that too many connections make the clusters merge.

Overall, these results show that the KNN-FN algorithm's performance depends strongly on how K is chosen.
The best K depends on the data type: small K values work well for complex and curved shapes (like 3-Spiral), while medium values work better for large or spread-out clusters (like Chainlink).
Choosing K around the elbow point, where ACC, ARI, and NMI become stable together, gives good and consistent clustering results while keeping the main structure of the data intact.

6. Conclusion

In conclusion, the proposed hybrid KNN-FN approach, which integrates KNN-Graph construction with the Fast Newman (FN) algorithm, provides an effective solution for clustering complex datasets with diverse shapes and densities.
By combining local neighborhood information from KNN with global community optimization through modularity maximization, the method successfully balances local cohesion and global separation.

Experimental results on both synthetic and real-world datasets show that KNN-FN consistently outperforms traditional clustering algorithms such as K-Means and Infomap across multiple metrics (ACC, ARI, and NMI).
However, the performance of the method remains sensitive to the choice of the neighborhood size k, and computational cost may increase for very large datasets.
Despite these challenges, KNN-FN demonstrates strong adaptability, making it a valuable tool for graph-based learning and clustering tasks that require capturing both local and global data relationships.
Future work will focus on automating parameter selection and improving scalability to handle high-dimensional and large-scale data more efficiently.

References

[1] Hassanzadeh, Tahereh, and Mohammad Reza Meybodi. "A new hybrid approach for data clustering using firefly algorithm and K-means." *The 16th CSI international symposium on artificial intelligence and signal processing (AISP 2012)*. IEEE, 2012.

[2] Rodriguez, Mayra Z., et al. "Clustering algorithms: A comparative approach." *PloS one* 14.1 (2019)

[3] Zhou, Bing, Bei Lu, and Salman Saeidlou. "A hybrid clustering method based on the several diverse basic clustering and meta-clustering aggregation technique." *Cybernetics and Systems* 55.1 (2024)

[4] Hajirahimi, Zahra, and Mehdi Khashei. "Hybrid structures in time series modeling and forecasting: A review." Engineering Applications of Artificial Intelligence 86 (2019)

[5] Uddin, S., Haque, I., Lu, H. *et al.* Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. *Sci Rep* **12**, 6256 (2022).

[6] Halder, R.K., Uddin, M.N., Uddin, M.A. *et al.* Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. *J Big Data* **11**, 113 (2024).

[7] Liu, Z., Barahona, M. Graph-based data clustering via multiscale community detection. *Appl Netw Sci* **5**, 3 (2020)

[8] Bhih, Amhmed, Princy Johnson, and Martin Randles. "An optimisation tool for robust community detection algorithms using content and topology information." *The Journal of Supercomputing* 76.1 (2020)

[9] Zhang, Xiao, et al. "A review of community detection algorithms based on modularity optimization." *Journal of Physics: Conference Series*. Vol. 1069. IOP Publishing, 2018.

[10] Newman, Mark EJ. "Fast algorithm for detecting community structure in networks." *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 69.6 (2004)

[11] Shi, Bing, Lixin Han, and Hong Yan. "Adaptive clustering algorithm based on KNN and density." *Pattern Recognition Letters* 104 (2018)

[12] Liu, Zijing, and Mauricio Barahona. "Graph-based data clustering via multiscale community detection." *Applied Network Science* 5.1 (2020)

[13] Li, Hao, et al. "A novel density-based clustering algorithm using nearest neighbor graph." *Pattern Recognition* 102 (2020)