IECO International Journal Of Industrial Electronics Control and Optimization
Print ISSN: 3517-2645
Online ISSN: 3568-2645
Homepage: https://ieco.usb.ac.ir/

Semi-Supervised Clustering with Improved Delaunay Graph Fusion and Pairwise Constraints

Shahin pourbahrami*
Assistant Professor, Department of Computer Engineering, Technical and Vocational University (TVU), Tehran, Iran, 5167759513, 09370445247, Corresponding author's email: shpourbahrami@tvu.ac.ir

Article Info

Article type:
Research Article

Article history:
Received: **********
Received in revised form: **********
Accepted: **********
Published online: **********

Keywords:
Delaunay graph, clustering, structural knowledge, pairwise constraints

ABSTRACT

In many data mining problems, leveraging structural and local connectivity information can significantly improve clustering performance. This paper presents a novel semi-supervised clustering framework that integrates weighted feature information, Delaunay-based graph construction, and pairwise constraints. First, feature weights are computed based on within-class pairwise variability, emphasizing dimensions that contribute most to local cluster structure. Weighted distances between samples are then calculated, and a Delaunay graph is constructed and filtered using an influence radius, preserving meaningful local geometric relationships while removing redundant edges. To capture higher-order neighborhood information, a GraphSAGE-style embedding propagates feature information through the graph, generating enriched low-dimensional representations of the data. Pairwise constraints are incorporated into the similarity matrix to encode prior knowledge about sample relationships, guiding the clustering process. Finally, semi-supervised clustering is performed using constraint-based spectral clustering. Experiments on benchmark datasets demonstrate that the combination of structural graph information, feature weighting, and pairwise constraints substantially improves clustering accuracy. The proposed framework is flexible and can be effectively applied across diverse data domains.

I. Introduction

In many data mining and machine learning tasks, clustering algorithms serve as essential tools for uncovering hidden structures within heterogeneous datasets [1]. Recent advances in this field have led to domain-specialized approaches tailored to a wide range of scientific and industrial applications. For instance, multi-scale graph convolutional networks (MS-GCNs) have shown strong performance in image clustering by leveraging both local and global structural information to generate more accurate and semantically meaningful groups [2]. In bioinformatics, clustering of single-cell RNA-Seq data has enabled the discovery of rare cell populations and complex gene-expression patterns, with modern methods combining deep autoencoders and graph convolutional architectures to preserve biological structure more effectively [3]. Clustering has also been applied to reduce data complexity and to more effectively model uncertainties in load and wind speed [4]. Similarly, contrastive learning–based clustering has improved image segmentation in medical and remote-sensing applications by creating more discriminative feature spaces [5]. Dynamic clustering of temporal networks has further enhanced the understanding of evolving communities and behavioral shifts in real-world social systems [6].

Semi-supervised clustering approaches have emerged in recent years, attempting to integrate prior knowledge (such as pairwise constraints) with unsupervised learning methods [7, 8]. On the other hand, using graph structures to model the geometric relationships of data has also proven to be an effective strategy for better representing data structure [9]. Among these, the Delaunay graph [10], as a geometric structure that naturally forms a network of relationships between points, has high potential for analyzing multidimensional data.

This study presents a novel framework for semi-supervised clustering that integrates a weighted Delaunay graph with structural knowledge (via feature weighting) and pairwise constraints. Using graph embedding techniques such as GraphSAGE-style embedding, the framework produces more meaningful numerical representations of the data. This method aims to improve clustering accuracy and robustness in handling complex and imbalanced datasets,

while offering strong generalizability across diverse data domains.

The structure of this paper is organized as follows: Section 2 reviews graph-based and geometry-based clustering algorithms, highlighting their strengths and limitations. Section 3 introduces the proposed algorithm that utilizes semi-supervised clustering with a combination of Delaunay graph and pairwise constraints. Section 4 presents and discusses experimental results, demonstrating the enhanced capability of the proposed method in identifying complex and nonlinear structures compared to traditional and advanced clustering techniques. Finally, Section 5 concludes the paper.

II. Related Work

Semi-supervised clustering via pairwise-constrained optimal graphs incorporates must-link and cannot-link constraints to optimize the graph structure [7]. A similarity graph is first constructed based on data points and then refined using pairwise constraints to better reflect the underlying cluster structure. Laplacian constraints preserve local graph connectivity, while a label propagation algorithm spreads supervision across nodes so that similar points are more likely to share the same label. This process produces cluster assignments that balance the internal data structure with limited external supervision.

Semi-supervised clustering via structural entropy introduces a method that integrates pairwise and label constraints into a structural entropy framework. The proposed algorithms support both flat and hierarchical clustering and have shown effectiveness in analyzing biological data such as single-cell RNA [8]. The approach constructs two graphs: a data graph based on similarity and a constraint graph encoding positive and negative relationships. A bi-dimensional structural entropy objective function, incorporating constraint consistency, enables simultaneous flat and hierarchical clustering.

Semi-supervised deep embedded clustering with pairwise constraints and subset allocation redefines the loss function based on sample similarity and introduces a subset assignment loss to improve performance on industrial textual data [11]. The method employs a similarity-based loss to better distinguish classes and a subset assignment loss to more effectively transfer knowledge from labeled data.

Semi-supervised clustering using pairwise constraints and local density structures is based on local density peaks and graph cuts. Pairwise constraints are used to segment inconsistent local trees, while the similarity matrix is refined with the E2CP algorithm [12]. The method incorporates intra- and inter-cluster conflict resolution and prevents tree fragmentation through root-changing and noise filtering. Finally, spectral clustering combined with E2CP further improves clustering accuracy.

"Some Methods for Classification and Analysis of Multivariate Observations" by J. MacQueen (1967) is a foundational work that introduced the *K*-Means algorithm. This iterative method partitions data into *k* clusters aiming to minimize the sum of squared distances from cluster centers. Though easy to implement and fast, it has drawbacks: sensitivity to initialization, inability to detect non-spherical clusters, poor noise handling, and requiring a pre-set number of clusters [13, 14].

Density-Based Spatial Clustering for Noisy Applications: This method introduces DBSCAN, a density-based algorithm to discover clusters in large spatial databases with noise. DBSCAN can detect clusters of arbitrary shapes and identify outliers without needing the number of clusters. However, it depends heavily on proper selection of parameters (ε neighborhood radius, *MinPts* minimum points), which can cause incorrect clustering or merging if misconfigured. Performance also declines sharply with variable-density data [15, 16].

Rodriguez and Laio introduced the DPC (Density Peak Clustering) algorithm, which discovers data clusters based on identifying points with high density and large distances from other dense points [17]. This method does not require specifying the number of clusters and intuitively detects cluster centers. However, its accuracy declines when facing complex structures and data with varying density. The study in [18] improves DPC using geodesic distance to identify clusters with more complex boundaries and uneven distributions. Despite this improvement, the new method remains sensitive to proper parameter selection, and computing geodesic distances can be time-consuming and costly in very large datasets.

A deep clustering approach extending *k*-means is proposed, where each cluster is represented by an auto encoder instead of a single centroid (*K*-DAE) [19]. Data points are assigned to the autoencoder with minimal reconstruction error, and clustering is optimized by minimizing the global reconstruction MSE across all auto encoders.

However, the performance of these algorithms, particularly with complex, noisy, and high-dimensional data, heavily depends on how similarity between samples is defined and on the intrinsic structure of the data [13, 15, 16]. One common limitation of traditional clustering methods is their neglect of structural knowledge. Yet in many real-world problems, valuable information about the relative importance of features or relationships between samples is available, which can significantly enhance clustering quality.

III. Methodology

A. Feature Weighting

In this stage, each feature in the dataset X ($X \in \square^{n \times d}$) is assigned a weight, which has n samples and d dimensions.

These weights are determined by analyzing the distances within the data points $d(x_i - y_j)$. The feature weights in Formula 1 are calculated.

$$w_k = 1/\sum_{k=1}^{d}\sqrt{\sum_{i\neq j=1}^{n} d(x_{ik} - y_{jk})^2} \qquad (1)$$

Where w_k is the weight of dimension (feature) *k*. A weight vector (feature weights) is defined whose size is equal to the number of features, and each element represents the importance of that feature in the distance calculations. In particular, if a larger weight is chosen for a feature, changes in that feature will have a greater effect on the overall distance and thus play a stronger role in clustering. This ensures that more significant features have greater influence on point connectivity and clustering. Finally, a feature receives a higher weight if the set of data points within that feature exhibits smaller within-class distances.

B. Combining the Delaunay Graph with Structural Knowledge

In the Delaunay graph, data points are connected to form triangles based on their geometric distribution, ensuring that no data point lies inside the circumcircle of any triangle [10]. This preserves the natural structure of the data. When certain features are known to be more important, this structural knowledge can be incorporated into the graph through feature weighting, thereby influencing distance calculations and the resulting embeddings to better reflect their importance.

C. Calculating Influence Radius and Filtering Edges Based on Circle Overlaps

For each data point, the influence radius defines how far its effect extends or, in other words, which neighboring points are within range. This process ensures only edges where both endpoints are meaningful neighbors are retained. Unlike a pure Delaunay graph that retains all triangle edges, this approach retains only structurally significant edges. For each point in the Delaunay graph set, the distance to the *k*th nearest neighbor is taken as the radius of influence for that point (r_{x_i}). In this way, the radius of influence of each point is determined based on the distance to several of its nearest neighbors and provides a measure of the scope of influence or local connections of that point. In Formula 2, edges whose length is greater than the sum of the radii of the two connected points are removed. This leaves only edges that indicate the true proximity between the two points.

$$keep \quad edge(x_i, y_j) \quad if \quad d(x_i, y_j) < r_{x_i} + r_{x_j} \qquad (2)$$

A similarity matrix is then constructed from the resulting graph connections (showing the relationship of each point to its neighbors in an adjacency matrix). Our innovation is that we improve the Delaunay graph connection method to preserve meaningful neighborhoods with a new and simple definition of neighborhood radius. We also preserve meaningful relationships and eliminate unlikely neighborhoods by embedding the Delaunay graph structure with the GraphSAGE algorithm.

D. Modifying the Similarity Matrix Using Pairwise Constraints [8] (Must-link and Cannot-link)

Suppose there is an initial similarity matrix $S \in R^{n\times n}$ built from the Delaunay graph. This matrix is then adjusted using constraints:

Must-link: For each pair *(i, j)* that must belong to the same cluster, their similarity is set to 1 (Formula 3).

$$S_{ij} = S_{ji} = \max(S_{ij}, \alpha) \qquad \alpha = 1 \qquad (3)$$

Must-link constraints ensure that certain points stay together, even if they appear far apart in feature space.

Cannot-link: Prevents incorrect grouping by setting the similarity of such pairs to zero or a very low value (Formula 4).

$$S_{ij} = S_{ji} = 0 \qquad (4)$$

E. Creating the Embedded Delaunay Graph

Instead of working directly on raw data, graph embedding algorithms such as Node2Vec and GraphSAGE are used to generate a numerical (embedded) representation that models the Delaunay graph structure more effectively. These embeddings are vectors that combine both the original features and the graph structure.

Node2Vec and GraphSAGE are graph embedding algorithms that aim to transform a graph into a vector space, preserving the structural properties of the graph and the similarity between nodes in this vector space. Each node in the graph (e.g., a data instance) is mapped to a numerical vector in a multidimensional space. Nodes that have similar or close structures in the graph will have close vectors. This feature makes clustering on vectors more accurate and flexible. Combining a weighted Delaunay graph with GraphSAGE allows preserving the structural and neighborhood information of the graph. As a result, the clustering algorithm becomes more accurate and robust to noisy points.

GraphSAGE is a representation learning method for graphs that, instead of looking at the entire graph, describes each node by looking at its neighbors. The basic idea is that for each node *v*, first the features of itself and its neighbors are collected, then they are summarized with an aggregation function (e.g., mean or max), and finally this summary is combined with the node's own features to create a new vector. This process is repeated in multiple layers so that the

node gets information not only from its immediate neighbors, but also from its more distant neighbors. In simple terms, GraphSAGE means “each node builds a summary of itself and its neighbors” to finally obtain a meaningful vector of its position in the entire graph. We used this method in the implementation of the proposed method due to its simplicity and locality to reduce execution time.

F. Constraint-Guided Spectral Clustering

Spectral clustering constructs a graph using the similarity matrix between samples, then uses eigenvalues and eigenvectors to embed the data into a new space where clustering is performed.

Overall, applying structural knowledge during clustering or data modeling offers significant advantages: First, assigning greater weights to key features can enhance clustering and classification accuracy while effectively reducing the influence of noise and irrelevant features. Second, the natural structure of the data is preserved, and more meaningful relationships between samples are identified, which is reflected in the quality of the similarity graph. Third, algorithms that leverage such knowledge show greater robustness when handling imbalanced and complex data structures. Additionally, this approach offers high flexibility, making it easily adaptable to various domains and data types. As a result, by incorporating feature importance into the graph structure, the overall clustering quality is significantly improved, especially in the presence of noise and high-dimensional data.

Figure 1 presents the pseudocode of the proposed clustering method. The algorithm begins by normalizing the input dataset and computing feature weights based on intra-class pairwise differences, ensuring that more discriminative features contribute more to the clustering process. Next, an influence radius is calculated for each data point, and edges are constructed using Delaunay triangulation with filtering, with a fallback to k-nearest neighbors if necessary. The resulting graph is transformed into an adjacency structure, and node embeddings are iteratively refined through a simplified GraphSAGE aggregation scheme. Finally, spectral clustering is applied to the learned embeddings, and the results are evaluated using Adjusted Rand Index and clustering accuracy, providing a robust and interpretable assessment of clustering performance.

Algorithm: Clustering with Improved Delaunay Graph Fusion and Pairwise Constraints

Input:
Dataset X (n × d), Labels y (for evaluation),
Number of clusters K, Neighbor parameter k

Output:
Cluster labels, ARI score, Clustering Accuracy

1. Normalize features of X using Min-Max scaling
2. Compute feature weights:
For each feature j:
For each class c:
Compute mean pairwise squared difference within class
Aggregate across classes, weighted by class size
Invert and normalize to obtain weight w_j
3. Apply feature weights to X → X_weighted
4. Compute influence radius for each point:
r_{x_i} = distance to k-th nearest neighbor
5. Build edges:
Try Delaunay triangulation
keep $edge(x_i, y_j)$ if $d(x_i, y_j) < r_{x_i} + r_{x_j}$
If fails, fallback to symmetric kNN graph
6. Construct adjacency dictionary from edges
7. Compute embeddings using simplified GraphSAGE:
Initialize embeddings = X_weighted
For T iterations:
For each node i:
Aggregate mean of neighbor embeddings
Update embedding: average of self and neighbors
8. Apply Spectral Clustering on embeddings → predicted labels
9. Evaluate results:
- Adjusted Rand Index (ARI)
- Clustering Accuracy (via Hungarian matching)

Return: predicted labels, ARI, Accuracy, number of edges

Fig.1. Pseudocode of the proposed method.

IV. Results

A. Clustering Evaluation Metrics

In evaluating the performance of clustering algorithms, three important metrics are used: ARI (Adjusted Rand Index), NMI (Normalized Mutual Information), and Accuracy. The ARI measures the degree of agreement between the predicted clusters and the actual labels, adjusting for the possibility of random agreement. It provides a score between -1 and 1, where 1 indicates perfect agreement. NMI measures the shared information between the predicted and true labels, normalized by entropy, and

provides a score between 0 (irrelevant) and 1 (perfect match) [19]. Accuracy (ACC) indicates the percentage of labels correctly matched by the clustering algorithm to the actual labels, assuming an optimal mapping between clusters and classes. The combination of these three metrics offers a comprehensive view of the quality of data partitioning.

B. Data Sets and the Hyperparameters

Table I presents a mix of synthetic and real-world datasets used for evaluating clustering algorithms. The columns in this table represent, respectively: data type (TYPE), number of clusters (*M*), number of samples (*N*), number of features or dimensions (*D*), and dataset name.

The synthetic datasets include six well-known examples: Blobs with 3 clusters and 300 samples in 2D space, Moons and Circles with 2 clusters and the same number of samples and dimensions, and three more complex sets—Three Densities, Anisotropic, and WingNut Horizontal—each with its own unique structure.

The real datasets include: Iris (3 clusters, 150 samples, 4 features), Wine (3 clusters, 178 samples, 13 features), Digits (10 clusters, 1797 samples, 64 features), Breast Cancer (2 clusters, 569 samples, 30 features), Diabetes (2 clusters, 768 samples, 8 features), USPS (10 clusters, 9298 samples, 256 features), Fashion-MNIST (10 clusters, 10000 samples, 784 features), and Letter (10 clusters, 10000 samples, 784 features). This diverse dataset design enables clustering algorithms to be evaluated under varying conditions—from simple 2D clusters to high-dimensional, real-world data—enhancing the assessment of their generalizability.

TABLE I DATASETS

DATASET	M	N	D	TYPE
Blobs	3	300	2	SYNTHETIC
Moons	2	300	2	SYNTHETIC
Circles	2	300	2	SYNTHETIC
Three Densities	3	300	2	SYNTHETIC
Anisotropic	3	300	2	SYNTHETIC
WingNut Horizontal	2	300	2	SYNTHETIC
Iris	3	150	4	REAL
Wine	3	178	13	REAL
Digits	10	1797	64	REAL
Breast Cancer	2	569	30	REAL
Diabetes	2	768	8	REAL
USPS	۱۰	۹۲۹۸	۲۵۶	REAL
Fashion-MNIST	۱۰	۱۰۰۰۰	۷۸۴	REAL
Letter	۱۰	۱۰۰۰۰	۷۸۴	REAL

Table II HYPERPARAMETERS USED FOR BASELINE METHODS

Dataset	K-Means (Clusters)	DBSCAN (ε)	DBSCAN (*minPts*)	DPC (Cut-off distance)
Circles	2	0.15	4	Data-driven
Three Densities	3	0.20	5	Data-driven
Blobs	3	0.25	5	Data-driven
Moons	2	0.18	4	Data-driven
Anisotropic	3	0.25	5	Data-driven
WingNut Horizontal	2	–	–	Data-driven
IRIS	3	0.60	4	Data-driven
WINE	3	0.65	6	Data-driven
DIGITS	10	1.50	10	Data-driven
Breast Cancer	2	0.70	5	Data-driven
Diabetes	2	0.50	5	Data-driven

To ensure fair comparisons across methods, the hyperparameters of the baseline algorithms were systematically tuned for each dataset in Table II. For *K*-Means, the number of clusters was set equal to the ground-truth number of classes. In DBSCAN, ε (the neighborhood radius) was determined using the *k*-distance graph heuristic, while *minPts* (minimum number of points) was selected according to common values in the literature, typically ranging from 4 to 10 depending on data dimensionality. For DPC, the cut-off distance and decision thresholds were chosen in a data-driven manner following the guidelines of the original implementation [13-18]. This careful tuning ensured that each baseline method operated under its best configuration, allowing an unbiased evaluation of the proposed method.

C. Clustering Results on the Datasets

Table III shows the ARI values for the synthetic datasets: the proposed method demonstrates flawless performance with ARI = 1.000. Meanwhile, DPC also provides near-perfect results, with most scores ranging between 0.990 and 1.000.

K-Means performs very well on spherical clusters such as Blobs and Circles, but shows a noticeable decline in performance on non-spherical shapes like Moons and Anisotropic (for example, Moons: ARI = 0.800). The DBSCAN algorithm performs worse compared to other methods, especially on structures with varying numbers of clusters and density distributions such as Three Densities and Anisotropic, where its ARI drops as low as 0.600.

In the WingNut Horizontal dataset, which has a unique geometric complexity, only the proposed method successfully achieved complete clustering, while other algorithms produced no valid results.

These results demonstrate the superiority of the proposed method and the relative stability of DPC compared to K-Means and DBSCAN in identifying nonlinear and complex clusters.

Table III ARI ON SYNTHETIC DATASETS

Dataset	Proposed METHOD	DPC	K-Means	DBSCAN
Circles	1.000	1.000	0.990	0.800
Three Densities	1.000	0.995	0.970	0.650
Blobs	1.000	0.995	0.980	0.920
Moons	1.000	1.000	0.800	0.700
Anisotropic	1.000	0.990	0.860	0.600
WingNut Horizontal	1.000	—	—	—

The results in Table IV show that the proposed algorithm performed extremely well on all synthetic datasets in terms of the NMI metric, achieving a value of 1.000 in all cases, indicating complete alignment of the clustering with the true data structure. In comparison, the DPC algorithm performed well on all datasets and, in most cases, was close to the proposed algorithm—e.g., 0.990 on Blobs and 0.980 on Circles—but did not reach the perfect score of 1.000. The *K*-Means algorithm performed worse, especially on non-spherical data such as Circles and Moons, which have more complex structures (with values of 0.000 for Circles and 0.395 for Moons, respectively). The DBSCAN algorithm also performed excellently on some datasets such as Circles and Moons (NMI = 1.000), but showed weaker performance on others such as Blobs (0.424) and WingNut Horizontal (0.669). Overall, the proposed algorithm consistently achieved the best performance across all datasets

Table IV NMI ON SYNTHETIC DATASETS

DATASET	PROPOSED METHOD	DPC	K-MEANS	DBSCAN
CIRCLES	1.000	1.000	0.990	0.800
THREE DENSITIES	1.000	0.995	0.970	0.650
BLOBS	1.000	0.995	0.980	0.920
MOONS	1.000	1.000	0.800	0.700
ANISOTROPIC	1.000	0.990	0.860	0.600
WINGNUT HORIZONTAL	1.000	—	—	—

Table V shows that the proposed algorithm outperformed other algorithms on most real-world datasets. For the Iris dataset, the proposed algorithm achieved an accuracy of 0.960, higher than those of DPC (0.889), K-Means (0.895), and DBSCAN (0.660). In the Wine dataset, although DPC had a higher accuracy (0.786), the proposed algorithm outperformed DBSCAN (0.365) and *K*-Means (0.698). On the Digits dataset, the proposed algorithm achieved an accuracy of 0.605 compared to DBSCAN (0.102) and *K*-Means (0.552), although the result for DPC was not reported. On the Breast Cancer dataset, the algorithm achieved 0.861, higher than DPC (0.589) and DBSCAN (0.620), and the accuracy for *K*-Means was not reported. Only in the Diabetes dataset did all algorithms perform poorly, with the proposed algorithm (0.405) scoring lower than DBSCAN (0.420) and lower than DPC (0.502). Overall, the proposed algorithm achieved the highest accuracy on most datasets, including challenging ones like Digits and Wine.

Table VI presents a comparative analysis of clustering performance across four datasets—Digits, USPS, Fashion-MNIST, and Letter—using the proposed method and baseline algorithms (*K*-DAE, *K*-MEANS, and DPC), measured by both NMI and ACC.

From the table, it is evident that the proposed method consistently achieves strong performance across all datasets. For the Digits dataset, it attains the highest NMI (0.83) and competitive ACC (0.86), slightly lower than *K*-MEANS in ACC but significantly outperforming *K*-DAE and DPC in both metrics. On the USPS dataset, the proposed approach outperforms all baselines, demonstrating its ability to capture complex structures and preserve cluster consistency (NMI = 0.77, ACC = 0.81). For Fashion-MNIST, although *K*-DAE

achieves slightly higher NMI (0.65 vs. 0.63), the proposed method surpasses all others in ACC (0.67), indicating a better alignment of predicted clusters with true labels. In the Letter dataset, the proposed method shows clear advantages over DPC and *K*-MEANS in both metrics, and the absence of *K*-DAE results highlights its limitation on this dataset.

Overall, Table VI demonstrates that the proposed semi-supervised clustering framework achieves a balanced performance in terms of NMI and ACC, offering robustness across different types of data, and outperforming traditional methods in most cases. Its consistent superiority in capturing meaningful cluster structures, combined with adaptability to various dataset complexities, underscores its effectiveness as a flexible and reliable clustering approach.

Table V ACCURACY ON REAL DATASETS

DATASET	PROPOSED METHOD	DPC	K-MEANS	DBSCAN
IRIS	0.960	0.889	0.895	0.660
WINE	0.657	۰,۷۸٦	0.698	0.365
DIGITS	0.605	–	0.552	۰,۱۰۲
BREAST CANCER	0.861	۰,٥۸۹	–	0.620
DIABETES	0.405	۰,٥۰۲	–	0.420

Table VI NMI and ACCC ON REAL DATASETS

DATASET	PROPOSED METHOD (NMI / ACC)	K-DAE (NMI / ACC)	K-MEANS (NMI / ACC)	DPC (NMI / ACC)
DIGITS	0.83 / 0.86	0.53 / 0.55	0.82 / 0.87	0.74 / 0.79
USPS	0.77 / 0.81	0.72 / 0.74	0.75 / 0.79	0.50 / 0.55
FASHION-MNIST	0.63 / 0.67	0.65 / 0.60	0.61 / 0.65	0.51 / 0.40
LETTER	0.72 / 0.75	- / -	0.70 / 0.74	0.53 / 0.44

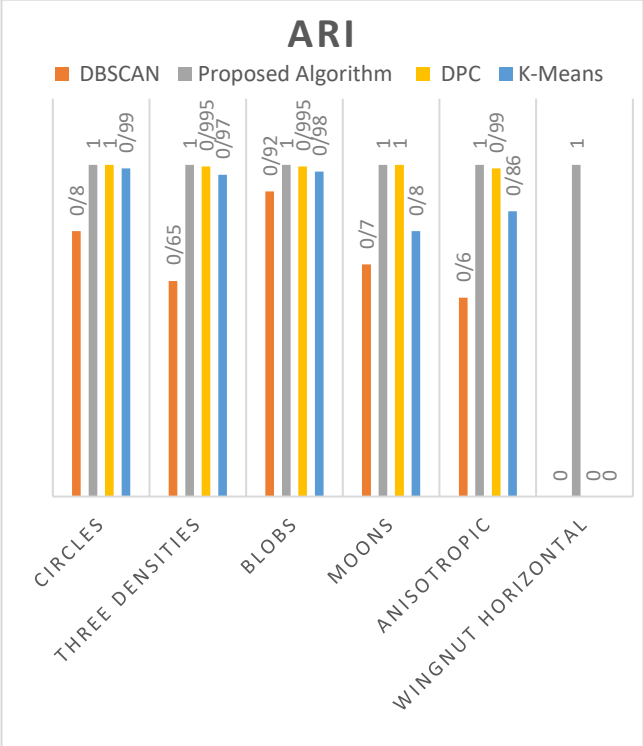

Fig 2: Comparison of the with three well-known clustering algorithms using metric: ARI

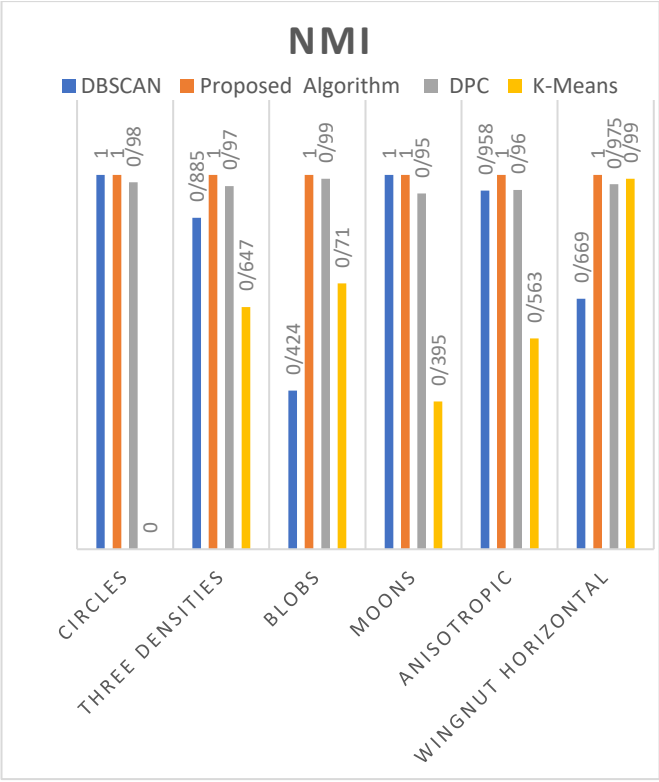

Fig 3: Comparison of the with three well-known clustering algorithms using metric: NMI

Figures 2 and 3 illustrate the performance comparison of four clustering algorithms (the proposed method, DPC, *K*-Means, and DBSCAN) on synthetic data using two metrics: ARI and NMI. The results show that the proposed method achieves values near 1.000 for both ARI and NMI across datasets, indicating very high clustering precision and consistency. The DPC algorithm also performs reasonably well but is slightly weaker than the proposed method in some cases, such as WingNut and Moons. In contrast, K-Means—and especially DBSCAN—suffer significant performance drops on complex datasets such as Circles and Three Densities; DBSCAN, in particular, produces very low results on some datasets even falling below 0.5. Overall, the charts demonstrate that the proposed method outperforms the other algorithms in terms of stability and accuracy when handling various cluster structures.

The fundamental differences between the proposed method and traditional algorithms (*K*-Means, DPC, DBSCAN) lie in cluster definition and data interpretation. *K*-Means is a centroid-based algorithm that assumes spherical clusters with similar variance, aiming to minimize the sum of distances from points to cluster centers. While simple and efficient, it struggles with non-spherical clusters, varying densities, or noisy data. DPC is a density-based approach that identifies cluster centers as points with high local density that are distant from other high-density points. Its performance depends heavily on parameters like cutoff distance and can be unstable in noisy datasets. DBSCAN also relies on density but focuses on connectivity, defining clusters as regions of densely connected points. Although it can handle arbitrarily shaped clusters and noise, it is sensitive to parameters such as neighborhood radius and *MinPts*, which may cause failures on datasets with variable densities.

In contrast, the proposed method integrates semi-supervised clustering with Delaunay-based graph fusion and graph embedding. This approach captures the underlying geometric and topological structure of the data. Feature weighting highlights meaningful dimensions, while influence radius filtering removes redundant edges to create a sparser, structurally significant graph. The GraphSAGE-style embedding propagates local neighborhood information, followed by spectral clustering to obtain globally consistent clusters. Unlike *K*-Means, it does not assume spherical clusters; unlike DPC and DBSCAN, it combines topological structure with semi-supervised guidance rather than relying solely on local density. This philosophy allows the proposed method to handle complex, non-linear cluster structures, improve robustness, and scale effectively to larger datasets, providing a more general and flexible framework for clustering compared to traditional methods.

D. Paired t-test Analysis

To assess the statistical significance of the proposed method's performance compared to other algorithms, a paired t-test was conducted between the results of the proposed approach and those of DPC, *K*-Means, and DBSCAN.

The results indicated that the comparison with DPC did not reveal any significant difference ($t = 0.25$, $p \approx 0.81$, $N = 9$), suggesting that the two methods perform very similarly. In comparison with *K*-Means, a relative difference was observed ($t = 1.91$, $p \approx 0.098$, $N = 8$), which, although not significant at the 95% confidence level, can be considered indicative of an advantage of the proposed method over *K*-Means at the 90% confidence level. By contrast, the comparison with DBSCAN showed a highly significant difference ($t = 5.49$, $p < 0.001$, $N = 10$), demonstrating the clear superiority of the proposed method in extracting more precise and stable data structures. These results confirm that the proposed method provides substantial improvements under diverse data conditions, particularly when compared with DBSCAN, and can serve as a powerful alternative in big data applications.

E. Algorithm Complexity Analysis

The computational complexity of the proposed algorithm can be summarized as follows. Data normalization scales each feature to [0,1] with complexity $o(\mathrm{n}*\mathrm{d})$. Feature weight computation involves calculating mean squared pairwise differences within each class, resulting in $o(d*n\hat{}2)$ in the worst case. Influence radius computation requires a full n x n distance matrix and sorting to find the *k*-th nearest neighbor, with complexity $o(\mathrm{n}\hat{}2*\mathrm{d})$. Edge filtering using Delaunay has complexity $o(n\log n + n^{\lceil d/2 \rceil})$, while the *k*-NN fallback is $o(\mathrm{n}\hat{}2*\mathrm{d})$ for brute-force or $o(n\log n*d)$ with *KD*-Tree. Building the adjacency dictionary from edges has complexity $o(E)$, and the GraphSAGE aggregation over T iterations is $o(T*d*E)$. Finally, spectral clustering requires Eigen-decomposition, typically $o(\mathrm{n}\hat{}3)$ but can be reduced to $o(\mathrm{n}*\mathrm{k}\hat{}2+\mathrm{n}\hat{}2)$ using *k*-NN affinity. Overall, the algorithm is dominated by feature weighting, distance computations, and spectral clustering, leading to approximate complexity o(d*n^2+T*d*E+n^3), which can be optimized for large datasets using sparse graphs and approximate methods.

F Contributions and Limitations of the Proposed Algorithm

Contributions: The proposed algorithm provides a systematic framework for semi-supervised clustering by

integrating feature weighting, influence-based graph construction, GraphSAGE-style aggregation. It accurately captures local connectivity and leverages weighted features to enhance clustering performance, particularly for datasets with complex structures. The use of influence radius filtering reduces redundant edges, resulting in sparser and more meaningful graphs. This sparsity improves efficiency and scalability. The GraphSAGE embedding allows information propagation across local neighborhoods, leading to more robust cluster representations.

Limitations: Despite its advantages, the algorithm has limitations. The feature weighting step requires computing pairwise distances within each class, which has $o(d*n\hat{}2)$ complexity and may become computationally expensive for large datasets. The influence radius and Delaunay graph construction also scale poorly with the number of samples and dimensions, potentially limiting applicability to high-dimensional or very large datasets without approximation or optimization. Finally, while the algorithm is robust for moderately sized and structured datasets, its performance on extremely sparse, noisy, or heterogeneous data may degrade without additional preprocessing or parameter adjustments.

V. Conclusions

This study presents an innovative semi-supervised clustering framework that integrates a structurally informed Delaunay graph, meaningful GraphSAGE-style embedding, and pairwise constraints to overcome the limitations of traditional methods in analyzing complex data. By weighting features according to their importance, the framework captures both local and global data structures. The embedding enables compact and learnable representations. Pairwise constraints guide the algorithm toward more logical cluster separation, especially in overlapping or imbalanced datasets, and spectral clustering on the modified similarity matrix maximizes the latent structure utilization. The approach improves accuracy, robustness to noise, and stability, while its design supports applications across diverse domains such as medical, biological, textual, and social network data. Despite computational challenges in feature weighting and graph construction for very large or high-dimensional datasets, as well as potential performance degradation on sparse or highly heterogeneous data, this method represents a significant step toward integrating domain knowledge and data structure to enhance semi-supervised clustering, providing a strong foundation for developing more intelligent models in the future.

REFERENCES

[١]Pourbahrami S, Balafar MA, Khanli LM, Kakarash ZA. A survey of neighborhood construction algorithms for clustering and classifying data points. Computer Science Review. 2020 Nov 1;38:100315, https://doi.org/10.1016/j.cosrev.2020.100315.

[٢]Xu Y, Huang D, Wang CD, Lai JH. Deep image clustering with contrastive learning and multi-scale graph convolutional networks. Pattern Recognition. 2024 Feb 1;146:110065, https://doi.org/10.1016/j.patcog.2023.110065.

[٣]Ren L, Wang J, Li W, Guo M, Yu G. Single-cell RNA-seq data clustering by deep information fusion. Briefings in Functional Genomics. 2024 Mar;23(2):128-37, https://doi.org/10.1093/bfgp/elad017.

[٤]Jadidoleslam M, Ghaseminejad M. Reliability-based Probabilistic Wind Power Planning Considering Correlation of Load and Wind. International Journal of Industrial Electronics Control and Optimization. 2022 Dec 1;5(4):304-15, https://doi.org/10.22111/ieco.2022.41531.1414.

[٥]Lv Z, Wu Z, Zhu J. Clustering-Guided Contrastive Prototype Learning: Towards Semi-Supervised Medical Image Segmentation. Pattern Recognition. 2025 Aug 23:112321, https://doi.org/10.1016/j.patcog.2025.112321.

[٦]You J, Hu C, Kamigaito H, Funakoshi K, Okumura M. Robust dynamic clustering for temporal networks. InProceedings of the 30th ACM International Conference on Information & Knowledge Management 2021 Oct 26 (pp. 2424-2433), https://doi.org/10.1016/j.jocs.2022.101877.

[٧]Nie F, Zhang H, Wang R, Li X. Semi-supervised clustering via pairwise constrained optimal graph. InProceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence 2021 Jan 7 (pp. 3160-3166).

[٨]Zeng G, Peng H, Li A, Liu Z, Yang R, Liu C, He L. Semi-supervised clustering via structural entropy with different constraints. InProceedings of the 2024 SIAM International Conference on Data Mining (SDM) 2024 (pp. 208-216). Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611978032.24.

[٩] Song Z, Yang X, Xu Z, King I. Graph-based semi-supervised learning: A comprehensive review. IEEE Transactions on Neural Networks and Learning Systems. 2022 Mar 18;34(11):8174-94, DOI: 10.1109/TNNLS.2022.3155478.

[١٠]Elshakhs YS, Deliparaschos KM, Charalambous T, Oliva G, Zolotas A. A comprehensive survey on Delaunay triangulation: applications, algorithms, and implementations over CPUs, GPUs, and FPGAs. IEEE Access. 2024 Jan 15;12:12562-85, DOI: 10.1109/ACCESS.2024.3354709.

[١١]Wang Y, Zou J, Wang K, Liu C, Yuan X. Semi-supervised deep embedded clustering with pairwise constraints and subset allocation. Neural Networks. 2023 Jul 1;164:310-22. https://doi.org/10.1016/j.neunet.2023.04.016

[١٢]Long Z, Gao Y, Meng H, Chen Y, Kou H. Semi-supervised clustering guided by pairwise constraints and local density structures. Pattern Recognition. 2024 Dec

1;156:110751. https://doi.org/10.1016/j.patcog.2024.110751

[١٣] McQueen, James B. "Some methods of classification and analysis of multivariate observations." *Proc. of 5th Berkeley Symposium on Math. Stat. and Prob.*. 1967.

[١٤] MacQueen J. Multivariate observations. InProceedings ofthe 5th Berkeley Symposium on Mathematical Statisticsand Probability 1967 (Vol. 1, pp. 281-297).

[١٥]Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. Inkdd 1996 Aug 2 (Vol. 96, No. 34, pp. 226-231).

[١٦]Xu X, Ester M, Kriegel HP, Sander J. A distribution-based clustering algorithm for mining in large spatial databases. InProceedings 14th International Conference on Data Engineering 1998 Feb 23 (pp. 324-331). IEEE. DOI: 10.1109/ICDE.1998.655795

[١٧]Rodriguez A, Laio A. Clustering by fast search and find of density peaks. science. 2014 Jun 27;344(6191):1492-6.

[١٨]Du M, Ding S, Xu X, Xue Y. Density peaks clustering using geodesic distances. International Journal of Machine Learning and Cybernetics. 2018 Aug;9(8):1335-49. DOI https://doi.org/10.1007/s13042-017-0648-x.

[19] Opochinsky, Yaniv, et al. "K-autoencoders deep clustering." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, DOI: 10.1109/ICASSP40776.2020.9053109.

Shahin Pourbahrami is currently an Assistant Professor in the Department of Computer Engineering at the Technical and Vocational University (TVU), Tehran, Iran. Her research interests include clustering algorithms, quantum computing, deep learning, and proteomics using deep learning approaches.