

IKCS: An Effective Intelligent K -Medoids Clustering Scheme Based on Granularity Rough Entropy

Linshu Chen^{1,2} , Liang Li¹ , Xueqi Ye¹ , Yuxiang Chen^{1,2*} , Wei Liang³ , Kuanching Li^{1,2*} 

¹ School of Computer Science and Engineering, Hunan University of Science and Technology (China)

² Hunan Provincial Engineering Research Center for Software Design and Services of Intelligent Connected Vehicles, Hunan University of Science and Technology (China)

³ The University of South China (China)

* Corresponding author: cheniyuxiang@hnust.edu.cn (Y. Chen), aliric@hnust.edu.cn (K. Li)

Received 3 August 2025 | Accepted 31 March 2026 | Published 12 August 2026



ABSTRACT

k -Medoids Clustering algorithm is one of the most commonly used clustering methods. However, it does not consider the weight of each feature, and the selection of center points is random, which may affect the clustering results. Granularity computing analyzes knowledge from various perspectives, considering the level of detail and completeness of that knowledge. This approach helps identify which features have the most significant impact on clustering outcomes, allowing for a focus on key features and the selection of more effective initial clustering centers. Based on this, we propose an Effective Intelligent k -Medoids Clustering Scheme Based on Granularity Rough Entropy (IKCS). Specifically, the granularity roughness entropy is first defined based on indistinguishable relationships and roughness, and feature weights are determined based on the roughness entropy. In addition, feature weight coefficients are introduced to explain the similarity function between samples, and they are integrated into k -Medoids. In the similarity calculation of clustering algorithms, the samples are subjected to coarse-grained clustering based on the above similarity function to obtain a set of coarse clusters. Subsequently, the algorithm uses the maximum minimum distance method to select initial cluster centers from the coarse clustering set. Experimental results on both artificial and well-known UCI datasets effectively demonstrate the superiority of the proposed IKCS. On artificial datasets with complex cluster distributions, IKCS achieves an Adjusted Rand Index (ARI) of 0.7582, outperforming the second-best algorithm (fast k -Medoids) by 8.1%. Meanwhile, running on 12 UCI datasets with high dimensionality, IKCS achieves an average accuracy that is 13.89% higher than the PAM algorithm and 7.19% higher than the FKM (fast k -Medoids) algorithm. Moreover, it shows strong performance in other key metrics, consistently achieving a lower clustering Sum of Squared Errors (SSE) and a higher average ARI compared to the baseline methods.

KEYWORDS

Clustering Algorithms, Granular Computing, Information Entropy, Rough Sets.

DOI: 10.9781/ijimai.2026.2266
eLocator: e2266

I. INTRODUCTION

CLUSTERING, as the core technology of unsupervised learning, is the fundamental step in data partitioning and pattern mining in modern data-driven scenarios. Its performance directly determines the effectiveness of downstream tasks such as IoT anomaly detection, medical image analysis, and large language model (LLM) preprocessing [1]. Over the past five years, the exponential growth of high-dimensional, noisy, and heterogeneous data has posed an urgent need for robust clustering algorithms. For example, in the field of medical diagnosis, the widely used breast cancer diagnosis dataset (WDBC) contains 30 dimensions of pathological characteristics; In the field of energy, modern smart grids are constantly generating massive (PB-level) sensor data streams, which urgently require efficient analysis

methods. According to relevant academic research, clustering is widely used in industrial artificial intelligence systems for data preprocessing, feature selection, and anomaly detection. However, traditional methods struggle to balance "accuracy" and "efficiency" when faced with complex data. Especially in scenarios such as anomaly detection in the Internet of Things (IoT), existing reviews have highlighted that traditional algorithms based on k -Means are limited in detection performance due to their sensitivity to initial center selection and their inability to effectively handle feature weights in high-dimensional data. In addition, this is largely due to unreasonable initial center selection and lack of consideration for weight in feature similarity calculation.

Existing clustering algorithms exhibit significant trade-offs when addressing three core challenges: initial cluster center selection, feature-weighted similarity measures, and computational complexity

Please cite this article as:

L. Chen, L. Li, X. Ye, Y. Chen, W. Liang, K. Li. IKCS: An Effective Intelligent k -Medoids Clustering Scheme Based on Granularity Rough Entropy, *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 10, no. 1, e2266, 2026, <http://doi.org/10.9781/ijimai.2026.2266>

TABLE I. COMPARISON OF DIFFERENT CLUSTERING ALGORITHMS

Model	Advantages	Disadvantages
PAM	robustness; noise resistance; distance measure	complexity; local optimal; initial point
Fast k-Medoids	efficiency; robustness; large-scale data.	initial point; local optimal; feature weight neglect
Mini-Batch k-Means	efficiency; resource conservation; large-scale data	low accuracy; initial point sensitive
G-Means	automatic clusters; normality; large-scale data.	initial point; complexity; non high-dimensional
DPC	automatic clusters; arbitrarily shaped data.	parameters sensitive; non high-dimensional
Spectral Clustering	automatic clusters; global Structure Snap.	parameters sensitive; non high-dimensional
DBSCAN	automatic clusters; arbitrarily shaped data.	parameters sensitive; non high-dimensional
GrC	fuzzy clustering; non-spherical; robustness	not intuitive; not for large-scale data.
Proposed IKCS	To improve k-Medoids by granularity rough entropy to better select initial centers.	

— limitations that have been repeatedly highlighted in mainstream research over the past five years. (1) Regarding initial center selection, while PAM [2], [3] and fast k -Medoids [4], [5] are widely used for their noise robustness, their random initialization methods lead to fluctuations in clustering accuracy exceeding 30% and frequent convergence to local optima. To mitigate this, Lucasius *et al.* [6] proposed a k -medoids algorithm based on a genetic algorithm to optimize initial centers via global search. Still, its $O(n^3)$ time complexity makes it unsuitable for large-scale datasets (e.g., 10^4 + samples). Xie *et al.* [7] introduced a k -medoids variant based on autoencoders for automatic initial center learning, which requires extensive labeled data (scarce in unsupervised scenarios) and fails on small-sample datasets like Soybean-small (47 samples). (2) In terms of similarity measures, most algorithms treat features with equal weight, ignoring the differing importance of various attributes in high-dimensional data. Tang *et al.* proposed the VWKFC algorithm [8], which adjusts feature contributions using weighted information granules; however, its kernel distance and density parameters rely on empirical thresholds, resulting in a 12% accuracy loss on non-spherically distributed datasets, such as Compound. Sun *et al.* proposed the FCPFS feature selection method based on fuzzy granular balls [9], relying on fuzzy mutual information and lacking a theoretical framework that integrates data granularity with clustering objectives. These limitations result in insufficient accuracy in similarity calculations, particularly in high-dimensional data (e.g., the 90-dimensional Libras Movement dataset), where redundant features can overshadow the impact of critical attributes. (3) Computational complexity is another critical bottleneck. Although distributed methods (such as Ng and Han [10] deploying k -medoids on the Spark framework) can reduce runtime for large-scale data, they require expensive cluster resources and suffer from reduced robustness in noisy environments. Density-based initialization methods accelerate center selection but fail to filter outliers—resulting in a center selection error rate exceeding 20% on datasets with 10% noise (like Haberman). Even multi-view or sparse methods (like the adaptive multi-granularity sparse subspace clustering proposed by Deng *et al.* [11]) only focus on dimensionality reduction rather than solving the inherent defects of k -medoids, failing to resolve the ‘complexity-robustness’ trade-off.

Clustering, as a crucial unsupervised learning method in data mining, has seen the emergence of various algorithms, such as k -Means, PAM, fast k -Medoids, Mini-Batch k -Means, G-Means [12], Density Peaks Clustering (DPC) [13], Spectral Clustering [14], DBSCAN [15], each of which has its own advantages and disadvantages as shown in Table I. The advantage of PAM lies in its use of the medoid (actual data point) as the cluster center, making it robust to noise and outliers, and not restricting it to a specific distance metric. In contrast to PAM, fast k -Medoids (k -Medoids) introduces a distance ratio parameter for selecting initial cluster centers, thereby minimizing the number of clustering iterations and achieving faster computation. Mini-Batch k -Means, a variant of k -Means, significantly improves efficiency on large-scale datasets by using small batches of data, though at the

potential cost of slightly reduced clustering accuracy. G-Means is an algorithm based on statistical testing; it utilizes a normality test to automatically determine the optimal number of clusters, but its disadvantage is a strict assumption about data distribution, making it unsuitable for non-normally distributed datasets. Furthermore, the DPC algorithm automatically identifies cluster centers by calculating the local density and distance of data points, making it suitable for finding clusters of arbitrary shapes and robust to noise. However, it performs poorly on high-dimensional data and has high computational complexity. Spectral Clustering utilizes the eigendecomposition of the graph Laplacian matrix to capture the global structure of the data and can handle complex-shaped clusters; however, it also suffers from high computational complexity and sensitivity to parameter selection. DBSCAN is a density-based algorithm that automatically identifies clusters and effectively handles noise, making it suitable for arbitrary shapes. However, it is sensitive to parameters and performs poorly on high-dimensional data or data with varying densities. In recent years, researchers have applied clustering algorithms to multiple domains, including intrusion detection, privacy protection, and social recommendation. Nevertheless, algorithms like PAM and fast k -Medoids are sensitive to the selection of initial points and may converge to a local optimum.

Granular Computing (GrC) [16], [17], as an emerging paradigm for processing complex data, has garnered significant attention over the past five years for its ability to decompose problems into manageable granules. Deng *et al.* [11] proposed a multi-granularity sparse subspace clustering method capable of adaptively adjusting data scales; however, this method focuses on subspace learning rather than optimizing k -medoids clustering. Xia *et al.* [18] designed a hierarchical clustering method for high-dimensional data based on GrC. Yet, it lacks mechanisms for initial center selection and feature weighting—failing to address the core deficiencies of k -medoids clustering. Bao *et al.* [19] explored dynamic granularity adjustment for data streams, though their method is specifically designed for time-series data and cannot process static high-dimensional datasets (such as the 33-dimensional Ionosphere dataset). These studies indicate that the potential of GrC to enhance k -medoids clustering remains underexplored. A critical knowledge gap exists in integrating GrC with k -medoids clustering to resolve the issues of random initial centers, the absence of feature weighting, and high computational complexity simultaneously.

It is worth noting that our research team has conducted preliminary explorations on weighted k -Medoids clustering in conference proceedings. Chen *et al.* [20] proposed a weighted k -Medoids method based on coarse entropy and attribute weights, which addresses the insufficient clustering performance of traditional k -Medoids in certain scenarios but still has limitations such as inadequate calculation of feature weights, a single type of experimental dataset, and a small number of evaluation metrics and comparative algorithms; Sun *et al.* [21] combined fuzzy quotient space theory with rough sets to develop an improved k -Medoids method, which resolves some performance constraints of traditional k -Medoids yet suffers from the lack of a new

measurement method, over-reliance on mature formulas, and limited experimental data and evaluation dimensions.

Therefore, based on the abovementioned observations and our previous work [22]–[26], an effective Intelligent k -Medoids Clustering Scheme (IKCS) based on granularity rough entropy is proposed to address these challenges systematically. The main contributions are as follows:

- **Definition of Granularity Rough Entropy (GRE):** A novel concept, "Granularity Rough Entropy" (GRE), is formally defined. This new measure integrates knowledge granularity with rough set theory to effectively quantify the uncertainty within data and the relative importance of different features.
- **GRE-Based Feature Weighting:** A novel feature weighting method is constructed based on the proposed GRE. This method assigns an objective weight to each feature, enhancing the differentiability of essential attributes, enabling the algorithm to move beyond equal-weighted similarity measures, thereby more accurately capturing the data's true local structure.
- **Intelligent Initial Center Selection:** A two-stage intelligent initialization strategy is designed to replace the random selection process. This strategy first utilizes the GRE-based feature weights to perform coarse-grained clustering and identify a high-quality candidate set. It then applies the maximum-minimum distance method to select well-separated and representative initial centers from this set, significantly improving stability.
- **The Proposed IKCS Scheme and Validation:** The complete IKCS is proposed, which integrates the GRE-based feature weighting and the intelligent initialization strategy into a unified framework. Extensive experiments on 12 UCI and 6 artificial datasets demonstrate that IKCS significantly outperforms 7 baseline algorithms in terms of clustering accuracy, robustness, and stability.

This work is necessary since traditional clustering algorithms, such as the fast k -Medoids algorithm, have limitations that can affect the bias in clustering results. Specifically, they are sensitive to the random selection of initial cluster centers and often fail to explain the different contributions of different features, treating all features as equally important, which may lead to unstable and inaccurate clustering results, especially when the initial centers (such as belonging to the same category or being very close to each other) are poorly selected. This article introduces a new method based on granular rough entropy to assign weights to features and optimize the selection of initial clustering centers, enhancing the distinguishability of features and improving the stability of the clustering process. It provides a more effective and feasible solution for complex data analysis tasks, offering a way to achieve more robust and accurate clustering.

The remainder of this work is organized as follows. Related work and definitions are presented in Section II, while the proposed IKCS, including its framework, feature weighting method, algorithm, and time complexity analysis, is detailed in Section III. The performances of IKCS on artificial and UCI datasets are analyzed in Section IV, and finally, the concluding remarks and future directions are presented in Section V.

II. RELATED WORK AND DEFINITIONS

In this section, we discuss related work on clustering, GrC, k -Medoids clustering, rough set model, and knowledge granularity. Meanwhile, we introduce essential concepts, including information table, indistinguishable relation, upper/lower approximations, and knowledge granularity, to facilitate an understanding of the proposed IKCS design. In Table II, we list the symbols used throughout this work.

TABLE II. DESCRIPTION OF SYMBOLS USED

Notation	Description
$[X_1, \dots, X_n]$	Rough Set
$ A $	The cardinality of set A
$IND(\{a\})$	Indistinguishable relationship to feature A
$U/IND(\{a\})$	Partitioning of $IND(\{a\})$
$\overline{Y_{A-a}}, \underline{Y_{A-a}}$	Upper and lower approximation of $IND(\{a\})$
$count_{zero}$	Number of features with importance of 0
$\rho_{A-a}(Y)$	Roughness of Y by the relation $IND(A-\{a\})$
$GK(A-\{a\})$	Knowledge granularity of features A
$GRE(\{a\})$	Granular coarse entropy of property a
$Sig_g(a)$	Importance of a 's coarse entropy
$weight_g(a)$	Weight of property a
$s(x_p, x_j)$	Similarity between samples a and b
$S(x_p, x_j)$	Similarity matrix
$S'(x_p, x_j)$	Fuzzy similarity matrix
AC	Accuracy of clustering
SSE	Sum of squared errors
ARI	Adjusted Rand Index

A. Literature Reviews on Clustering Algorithms

Clustering is a key data analysis method that divides a group of objects into multiple clusters based on specific criteria, so that objects within the same cluster have high similarity. In recent years, many innovative clustering algorithms [8], [9] have been proposed, and clustering has been widely applied in various fields, such as security of Internet of Vehicles (IoV) [27], [28], blockchain [29]–[31], embedded system [32], privacy protection [33]–[36], mobile robot machine learning [37], [38]. To address the noise problem in some multi-tag data, the FCPFS algorithm proposes a feature selection method based on fuzzy particle clusters and fuzzy mutual information. Through fuzzy particle pre-processing and fuzzy entropy calculation, the accuracy of feature selection and classification performance is significantly improved. The VWKFC algorithm enhances the initialization method of cluster centers by incorporating kernel distance, density-based viewpoint, and weight information granules, effectively reducing the impact of noise and outliers. It exhibits a good clustering effect and fast convergence performance in high-dimensional data processing.

The k -Medoids algorithm is a classical clustering method, renowned for its robustness to noise and outliers. However, its performance is highly dependent on the selection of the initial center, which has received extensive attention in recent years. Improper selection of the initial center may cause the algorithm to converge to a local optimum, thereby affecting the accuracy and stability of the clustering results. Recent research has shown that using heuristic methods or density-based strategies to initialize the center can significantly enhance the clustering effect. For example, Rodriguez and Laio [39] proposed an initialization method based on local density, which can effectively reduce the number of iterations and improve the quality of clustering. In addition, the sensitivity of k -Medoids to the initial center also raises questions about its scalability, especially when dealing with large-scale datasets. To solve this problem, Ng and Han [10] proposed an improved k -Medoids method based on distributed computing. By combining the algorithm with the Spark framework, their performance in the big data environment was significantly improved. In recent years, many improved methods have been proposed to optimize the performance of k -Medoids. For example, Lucasius *et al.* [6] proposed

an improved k -Medoids method based on a genetic algorithm, which significantly improved the stability of clustering results by optimizing the initial center selection through global search. In addition, Xie *et al.* [7] introduced a k -Medoids variant based on deep learning, which automatically learns the initial center using a self-encoder network, thus reducing the dependence on manual intervention. However, optimizing the initial center selection without sacrificing clustering quality remains a significant challenge in k -Medoids research.

Recent studies have also attempted to optimize clustering algorithms from various perspectives. Dhar *et al.* [40] proposed an entropy-based feature selection method as a preprocessing step, which addresses the problem of insufficient clustering data quality but still has limitations such as only optimizing the preprocessing stage, not improving the algorithm itself, and potential information loss; Amalia [41] proposed a method to improve k -Medoids using Z-Score initialization, which resolves the issues of excessive iterations and low efficiency of k -Medoids yet lacks consideration of feature weights and only involves a single optimization dimension; Additionally, Jiang *et al.* [42] introduced a fast Minimum Spanning Tree (MST) clustering algorithm based on Granular-Ball computing, which solves the problems of low efficiency and poor noise resistance of the MST algorithm when processing large-scale data but is specific to the MST algorithm, irrelevant to k -Medoids, and unable to address the core defects of k -Medoids.

As a new computing paradigm, GrC offers a fresh perspective on data analysis and pattern recognition by decomposing complex problems into subproblems of varying granularity. In recent years, the combination of GrC and cluster analysis has received extensive attention and shown significant advantages. The core idea of GrC is to capture the essential characteristics of data through multi-level and multi-granularity information processing, thereby improving the robustness and interpretability of clustering algorithms. For example, Chen *et al.* proposed a hierarchical clustering method based on GrC, which can effectively deal with noise and redundant information in high-dimensional data. In addition, GrC also supports dynamic granularity adjustment, allowing the clustering process to be optimized adaptively in response to changes in data distribution. This flexibility not only improves the accuracy of clustering results but also provides a more intuitive framework for interpreting complex datasets. Recent research further shows that combining GrC with traditional clustering algorithms (such as k -Medoids and DBSCAN) can significantly improve its performance on high-dimensional and nonlinear data. Therefore, GrC provides a powerful tool for clustering analysis, which can effectively deal with the challenges of complexity and diversity in modern data analysis.

B. k -Medoids Clustering

The medoids clustering algorithm is a partition-based clustering method with unique characteristics compared to the k -Means algorithm. k -Medoids clustering uses actual data samples as the center points of the cluster rather than simply using sample means, thereby enhancing the algorithm's robustness to noisy data and isolated points. The core idea of this algorithm is to randomly select a representative sample from the dataset as the center point for each cluster. Subsequently, based on the distance between other samples and these center points, they are assigned to the cluster corresponding to the nearest center point. Subsequently, clustering quality is discussed by repeatedly using non-representative objects to replace representative objects. Cluster quality is typically measured using a cost function that reflects the distance and allocation between objects within the cluster. When a central point is replaced by a non-central point, all other points are reassigned, which may improve clustering performance. The iterative process of the algorithm continues until there is no significant change

in clustering quality. The criterion function is as

$$J = \sum_{i=1}^k \sum_{j=1}^m d(O_j, C_i) \quad (1)$$

in which there are k class clusters, with the center points of each class cluster being $C_1, C_2, \dots, C_k$, and the objects contained in each cluster are $O_1, O_2, \dots, O_m$, $d(O_j, C_i)$ represents the distance from the object O_j to the center C_i of the class cluster.

Given the i^{th} class cluster contain N samples, which is respectively expressed as $x_{i1}, x_{i2}, \dots, x_{iN}$. Then, the class cluster m_i is defined as

$$m_i = \left\{ x_{ij} \mid \min_{j=1}^n |x_{ij} - \frac{1}{N} \sum_{k=1}^N x_{ik}| \right\} \quad (2)$$

The advantage of the k -Medoids clustering algorithm is that it takes the actual data object closest to the cluster center as the center of the class, resulting in a highly robust algorithm, especially when dealing with small datasets. However, k -medoids algorithms have specific issues with randomly selecting initial clustering centers, which may result in varying clustering results for different initial centers. Moreover, the algorithm may slowly converge, making it less suitable for processing large datasets.

C. Rough Set Model

GrC is typically used to handle complex problems, where the details and complexity of the problem may make it difficult for traditional methods to solve effectively. The core idea is to decompose the problem into different levels or granularity based on its characteristics and requirements and handle it at each granularity level [43]. GrC has been widely used in many fields, including computer vision, information retrieval and recommendation system, complex problem solving, fuzzy system and fuzzy logic, pattern recognition, knowledge discovery and management, edge computing [44], intelligent control, artificial neural network [45], natural language processing, and many more. Notably, many scholars have recently focused on improving new clustering methods based on GrC.

The rough set model, being one of the GrC models, is a mathematical tool and theoretical framework used to process complex data and help reveal patterns and knowledge within the data. First proposed in 1982 by Polish scholar Zdzislaw Pawlak, it addresses the incompleteness, uncertainty, and ambiguity of data to understand the data's inherent structure effectively [43], [46]. The core idea of rough set theory is to divide the dataset into distinct equivalent classes, capturing the inherent relationships and patterns within the data. This theory is applied in multiple fields, including feature selection, knowledge discovery, and decision support. The theory, as mentioned above, effectively handles complex datasets with uncertainty and missing information. Researchers and analysts can efficiently understand data and discover hidden patterns and valuable information through rough-set theory. This theory provides powerful tools for data science and information processing and is essential in modern data analysis.

Two essential concepts in rough sets are the indistinguishable relation and B-upper/B-lower approximations. These concepts will serve as the basis for the clustering method proposed in this work.

Given an information table [17] $IS = (U, A, V, f)$, for any sub-features $B \subseteq A$, sub-objects $X \subseteq U$ and $(x, y) \in U \times U$, then the indistinguishable relation $IND(B)$ determined by B is defined as

$$IND(B) = \{ \forall a \in B (f(x, a) = f(y, a)) \} \quad (3)$$

and the B -upper approximations and B -lower approximation of X are defined as

$$\overline{X}_B = \cup \{ [x]_B \in U/IND(B) : [x]_B \cap X \neq \emptyset \} \quad (4)$$

$$\underline{X}_B = \cup \{ [x]_B \in U/IND(B) : [x]_B \subseteq X \} \quad (5)$$

Indistinguishable relation $IND(B)$ is an equivalence relation on the domain U , which is reflexive, transitive, and symmetrical. It divides U into disjoint equivalence classes, and the set of all equivalence classes forms a division of U , denoted $U/IND(B)$, which can be regarded as a piece of knowledge on the domain U . An equivalence class is a collection of objects with the same feature values, which is the most concise characterization of the domain U without any redundant information.

$\overline{X_B} \neq X_B$ indicates that the object subset X cannot be precisely divided by the equivalent class in $U/IND(B)$, if it cannot be fully represented by known information, X is a rough set. The B -lower approximation of X contains all elements that can be classified into X using knowledge $U/IND(B)$, and the B -upper approximation of X contains all elements that may belong to X .

D. Knowledge Granularity

Knowledge is believed to be related to this equivalence relation, and the equivalence class produced by the equivalence relation demonstrates that knowledge has granularity, as further developed in rough set theory. Therefore, researchers worldwide represent a similar granular structure of knowledge through equivalence classes and relations, and calculate the size of the knowledge granularity represented by equivalence relations and classes.

Knowledge granularity is essential in data processing and clustering tasks with specific applications, including feature selection and dimensionality reduction. We can better understand the relationship among the features of data and select the most informative features by introducing the concept of knowledge granularity. Additionally, knowledge granularity facilitates data pre-processing, including discretization, smoothing, and normalization, thereby reducing noise and uncertainty in the data and enhancing the accuracy of data analysis. In summary, the application of knowledge granularity can enhance the flexibility and effectiveness of data processing and clustering tasks, enabling a deeper understanding of the structure and internal relationships of data, thereby improving applications in specific fields, such as data mining, decision support, and model interpretation.

Given an information table [17] $IS = (U, A, V, f)$, where $U/IND(A) = \{X_1, X_2, \dots, X_m\}$, then the knowledge granularity of A is defined as

$$GK(A) = \frac{1}{|U|^2} \sum_{i=1}^m |X_i|^2 \quad (6)$$

In Eq.(6), $\sum_{i=1}^m |X_i|^2$ is the number of elements in the equivalence relationship determined by $U_{i=1}^m (X_i \times X_i)$, and $GK(A)$ is the sum of squares of the sets obtained through partitioning set U by the equivalence relation $IND(A)$, divided by the square of the magnitude of set U . If $K(A) = \varpi$, then the information granularity of A can be minimized by $|U|/|U|^2 = 1/|U|$. If $K(A) = \delta$ then the information granularity of A can be minimized by $|U|^2/|U|^2 = 1$.

III. THE PROPOSED IKCS

In this section, we present a comprehensive description of the proposed IKCS, including the overall IKCS framework, the feature weighting method based on rough entropy granularity, a detailed design of the IKCS algorithm, and an analysis of the algorithm.

A. IKCS Framework

The detailed architectural framework of the proposed IKCS is illustrated in Fig. 1, with its primary objectives outlined below.

- To calculate the feature weight, where granular entropy is calculated through the indiscernibility relation to determine feature weights.

- To select cluster centers, where the fuzzy similarity matrix is first obtained by using the weight of each feature, and then the initial clustering centers are selected based on the maximum–minimum distance.
- To design the IKCS algorithm, where the core point is the above-obtained feature weights, and then the boundary points are allocated until the convergence conditions are met.

B. Feature Weighting Method by Granularity Rough Entropy

In the classic k -Medoids clustering algorithm, all features are assigned the same weight, which means that the contribution of each feature to the clustering results is considered consistent. However, in practical applications, there are often significant differences in the importance of different features. Some features may play a crucial role in clustering results, while the influence of other features may be minimal or even negligible. Therefore, treating all features equally important may lead to inaccurate measurement of differences, affecting the quality of clustering results. By optimizing feature weights, we can improve the accuracy and effectiveness of clustering.

To solve this problem, this work proposes a feature weight calculating method, which introduces a new concept of granularity rough entropy by utilizing concepts of knowledge granularity and roughness in rough set theory. Therefore, before designing the feature weight calculation method in IKCS, three critical concepts – sample granularity, roughness, and granularity rough entropy – are newly defined as follows.

Given an information table $IS = (U, A, V, f)$, where $X \subseteq U$, $P, Q \subseteq A$, $U/P = \{Y_1, Y_2, \dots, Y_m\}$ and $U/Q = \{X_1, X_2, X_3, \dots, X_n\}$, then the roughness of X under $IND(P)$ is defined as

$$\rho_P(X) = \frac{|\overline{X_P} - X_P|}{|\overline{X_P}|} \quad (7)$$

and the granular coarse entropy $GRE(P, Q)$ of P under $IND(Q)$ is defined as

$$GRE(P, Q) = GK(Q) \times \sum_{i=1}^m \rho_Q(Y_i) \log_2(\rho_Q(Y_i) + 1) \quad (8)$$

In Algorithm 1, to evaluate the importance of an individual feature $o_i \in M$, we map the single feature $\{o_i\}$ to the set P , and the remaining feature set $M - \{o_i\}$ to the set Q . Consequently, the granular coarse entropy $S_{igC}(o_i)$ is calculated equivalently to $GRE(\{o_i\}, M - \{o_i\})$, ensuring consistency between the theoretical definitions and algorithmic implementation.

In the IKCS framework, calculating feature weights is of pivotal importance. The core step of the feature weight calculating method is designed with the following essential missions.

- The sample granularity $GK(A-a)$ is calculated based on Eq.(6), measuring the degree of knowledge representation of the dataset with feature a .
- The roughness $\rho_{A-a}(Y)$ is determined by Eq. (7) to evaluate the distinguishability of the property a , being added the temporary variable $temp$ to adjust its contribution using the logarithmic function.
- The granularity coarse entropy $GRE(\{a\}, A - \{a\})$ is quantified according to Eq.(8), measuring the contribution of feature a to the dataset.
- Most crucially, based on the above granularity rough entropy $GRE(P, Q)$, the feature weight of a in our proposed IKCS is defined as

$$\text{weight}_c(a) = \begin{cases} \frac{1}{2} \times \left(1 + \frac{\text{count}_{\text{zero}}}{|A| + \sqrt{|A| - \text{count}_{\text{zero}}}} \right), & \text{if } \text{Sig}_G(a) = 0 \\ 1 + \text{Sig}_G(a), & \text{if } \text{Sig}_G(a) > 0 \end{cases} \quad (9)$$

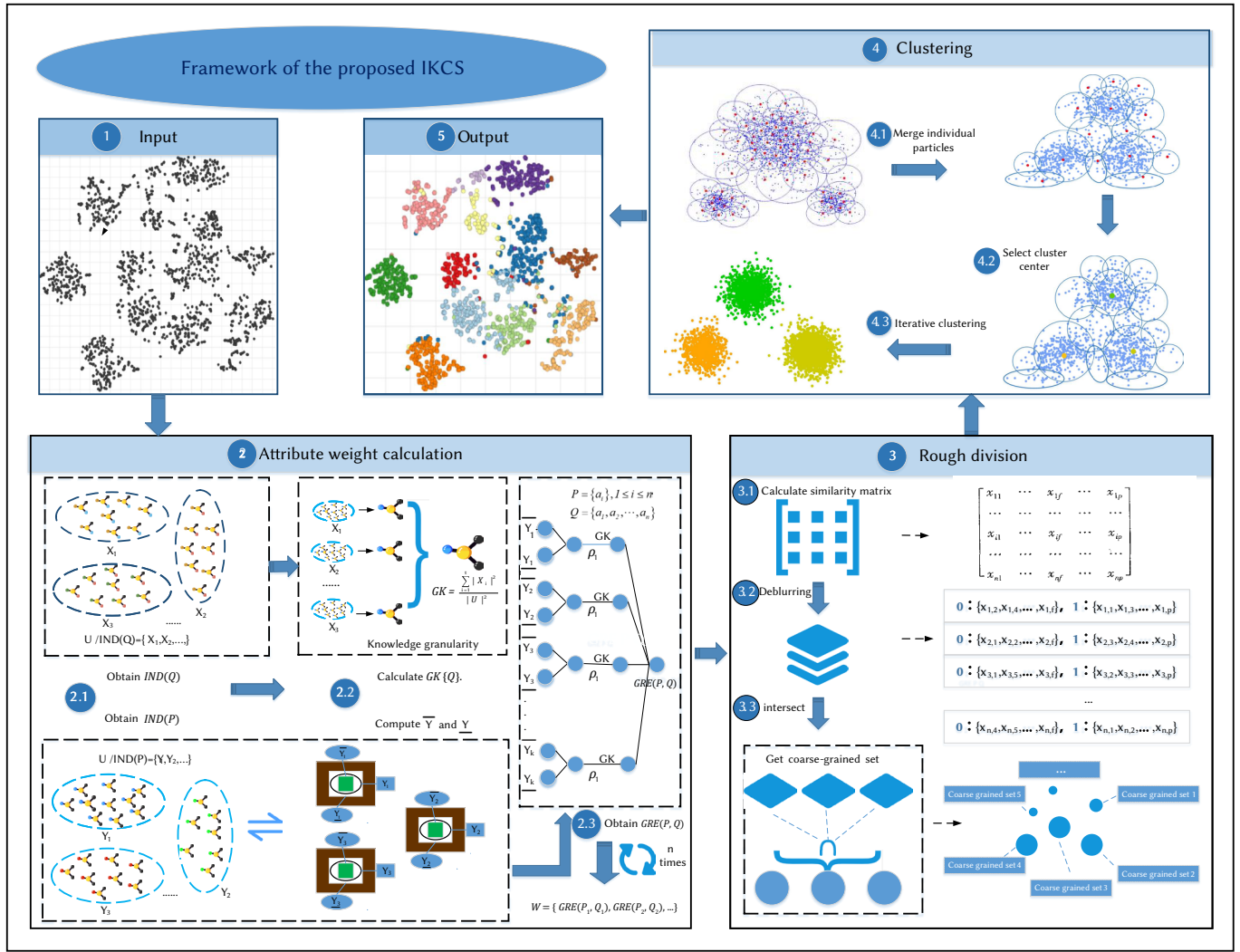


Fig. 1. Framework of the Proposed IKCS.

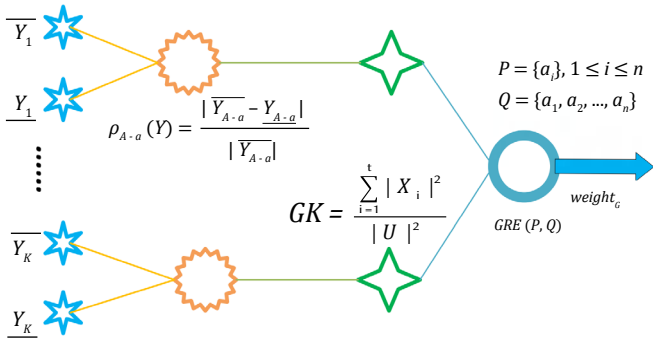


Fig. 2. The schematic diagram of feature weight calculation.

- (5) At last, the above feature weight w_k is introduced to redefine the new sample similarity of our proposed IKCS as follows

$$s(x_i, x_j) = \frac{1}{1 + \sum_{k=1}^{|M|} w_k |x_{ik} - x_{jk}|} \quad (10)$$

In Eq. (10), sample similarity is closely tied to clustering, as it often depends on similarity calculations to group similar data points into clusters that exhibit intrinsic similarities. Consequently, the outcomes of these similarity calculations have a significant impact on the clustering results. In this work, we first incorporate feature weights into the similarity calculations.

Taking into account the importance of various features. We reduce the weight of some less important features and eliminate those that are not significant for similarity calculations in the dataset.

The corresponding schematic diagram of the proposed feature weight calculating method using rough entropy is presented in Fig. 2, along with its detailed pseudo-code as shown in Algorithm 1.

In Algorithm 1, Line 2 obtains the indistinguishable relation $IND(M - o_i)$'s partition of the domain U by Eq. (3), and Line 3 achieves the indistinguishable relation $IND(M - \{o_i\})$'s partition of the domain U by Eq. (3). In fact, Line 2 and Line 3 will take the most running times of proposed IKCS, which will be Analyzed in the following. Line 4 calculates the sample granularity $GK(M - \{o_i\})$ by Eq. (6). Lines 6 calculates the upper approximation $\bar{Y}_{m, M - \{o_i\}}$ and lower approximation $\underline{Y}_{m, M - \{o_i\}}$ of Y_m by Eq. (4) and Eq. (5), concerning the relation $IND(M - \{o_i\})$. Line 7 calculates the granular roughness $\rho_{M - \{o_i\}}(Y_m)$ of Y_m relative to the relation $IND(M - \{o_i\})$ by using Eq. (7). Lines 10 obtains the granular rough entropy $Sig_G(o_i)$ using Eq. (8), and Line 12-16 calculates the weights according to Eq. (9). According to Eqs. (6), (7), and (8), the value of $Sig_G(o_i)$ is always ≥ 0 , so there is no need to handle situations where it is < 0 .

C. IKCS Model

The overall process of the proposed IKCS model is illustrated in Fig. 3, with its main processes outlined as follows.

Algorithm 1. The feature weight calculating method.

REQUIRE

 Properties $M = o_1, o_2, \dots, o_r$

 Samples set $U = u_1, u_2, \dots, u_n$

 Numbers of clusters k

ENSURE

 weighted feature $weight_G(M\{o_1\}), \dots, weight_G(M\{o_r\})$

PROCEDURE

```

1: for  $i \leftarrow 1$  to  $r$ 
2:    $U/IND(M-\{o_i\}) = u_1(M-\{o_i\}) \cap u_2(M-\{o_i\}) \cap \dots \cap u_n(M-\{o_i\}) = \{x_1, \dots, x_s\}$ 
3:    $U/IND(\{o_i\}) = u_1(\{o_i\}) \cap u_2(\{o_i\}) \cap \dots \cap u_n(\{o_i\}) = \{Y_1, \dots, Y_m\}$ 
4:    $GK(M-\{o_i\}) = \frac{\sum_{p=1}^s |x_p|^2}{|U|^2}$ 
5:   for  $j \leftarrow 1$  to  $m$ 
6:     Calculating  $\overline{Y_{m,M-\{o_i\}}}$  and  $\overline{Y_{m,M-\{o_i\}}}$ 
7:      $\rho_{M-\{o_i\}}(Y_m) = \frac{\overline{Y_{m,M-\{o_i\}}} - Y_{m,M-\{o_i\}}}{\overline{Y_{m,M-\{o_i\}}}}$ 
8:      $temp += \rho_{M-\{o_i\}}(Y_m) \log_2(\rho_{M-\{o_i\}}(Y_m) + 1)$ 
9:   endfor
10:   $Sig_G(o_i) = GK(M-\{o_i\}) \times temp$ 
11: endfor
12: if  $Sig_G(o_i) > 0$  then
13:   $weight_G(o_i) = 1 + Sig_G(o_i)$ 
14: else // The implicit condition here is  $Sig_G(o_i) = 0$ 
15:   $weight_G(o_i) = \frac{1}{2} \times \left(1 + \frac{\text{count\_zero}}{|M| + \sqrt{|M|} - \text{count\_zero}}\right)$ 
16: endif
17:  $weight_G(M\{o_1\}), \dots, weight_G(M\{o_r\})$ 
    
```

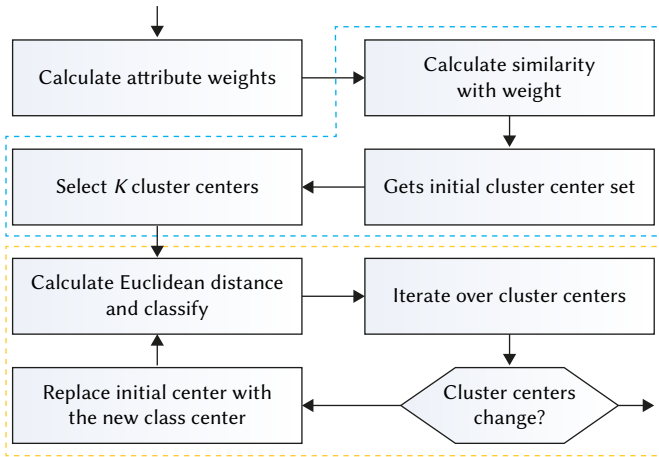


Fig. 3. The flow chart of the proposed IKCS.

- (1) The corresponding feature weight $weight_G(M\{o_i\})$ is obtained by calculating the granular coarse entropy in Algorithm 1.
- (2) Selecting initial cluster centers.

The similarity between the two samples x_i, x_j is computed to construct a similarity matrix $S(x_i, x_j)$. A threshold d , which is adaptively determined by the mean value of the similarity matrix S , is set. Then, fuzzification is applied to the similarity $S(x_i, x_j)$ to obtain a fuzzy similarity matrix $S'(x_i, x_j)$. An initial clustering is performed based on the fuzzy similarity matrix $S'(x_i, x_j)$, and each cluster's centroids are calculated.

This process results in forming a candidate set $Medoids = \{m_1, m_2, \dots, m_j\}$ for the initial cluster centers. The maximum-minimum distance method selects k cluster centers from the candidate set of initial cluster centers.

(3) Cluster Iteration.

Each sample is assigned to the nearest cluster center for the initial clustering results. After that, the sum of distances between each sample in each cluster and its cluster center is calculated. Updating the cluster centers involves selecting the object within each cluster whose sum of distances to other objects in the same cluster is the smallest. This object is chosen as the new cluster center, replacing the original center. If the sum of distances between all samples in each cluster and their respective cluster center remains unchanged for all clusters, then the iteration stops; otherwise, the cluster centers are updated.

Algorithm 2. The proposed IKCS algorithm.

REQUIRE

 Properties $M = o_1, o_2, \dots, o_r$

 Samples set $U = u_1, u_2, \dots, u_n$

 Numbers of clusters k

ENSURE

 Clustering results $C_1, C_2, \dots, C_k$

PROCEDURE

```

1: Calculating weighted features by Algorithm 1
    $weight_G(M\{o_1\}), \dots, weight_G(M\{o_r\})$ 
2: for  $i \leftarrow 1$  to  $n$ 
3:   for  $j \leftarrow 1$  to  $n$ 
4:      $s(x_i, x_j) = \frac{1}{1 + \sum_{k=1}^{|A|} w_k |x_{ik} - x_{jk}|}$ 
5:      $d, S'(i, j) = \begin{cases} 1, s(i, j) > d \\ 0, s(i, j) < d \end{cases}$ 
6:   endfor
7: endfor
8: Classifying the samples according to matrix  $S'(i, j)$  to obtain a coarse granule set  $= [X_1, \dots, X_t]$ 
9: for  $i \leftarrow 1$  to  $t$ 
10:   $m_i = \{x_{ij} | \min_{j=1}^n |x_{ij} - \frac{1}{N} \sum_{k=1}^N x_{ik}|\}$ 
11: endfor
12: Medoids  $= [m_1, \dots, m_t]$ 
13:  $d_1, d_2 = \min(s(m_i, m_j))$  where  $i \neq j$ 
14:  $d_k = \min \{ \max \{s(m_p, d_1), \dots, s(m_p, d_{k-1})\} \}$ 
15: do
16:   for  $i \leftarrow 1$  to  $n$ 
17:     for  $j \leftarrow 1$  to  $n$ 
18:       if  $u_j$  is close to  $d_i$  then
19:          $u_j \in C_i$ 
20:       endif
21:     endfor
22:   endfor
23:   Updating the cluster center.
24: until cluster center remains unchanged
    
```

The detailed pseudo-code is provided in Algorithm 2. Line 4 calculates the similarity between each pair of samples using Eq.(10). Line 5 sets a threshold value d . If $s(i, j) > d$, then $S'(i, j) = 1$. If $s(i, j) < d$, then $S'(i, j) = 0$. Line 8 classifies the samples according to matrix

$S'(i, j)$ to obtain a coarse granule set $= [X_1, \dots, X_k]$. Line 10 calculates the cluster centers of each coarse sample set using Eq. (2) and places the cluster centers of each coarse sample set into the cluster centre candidate set Medoids. Line 13 selects the two samples, m_i and m_j ($i \neq j$), with the minor similarity from the initial cluster center candidate set based on the similarity matrix $S(i, j)$, where $d_1 = m_i$ and $d_2 = m_j$, and then removes them from the initial candidate set. Line 14 uses the maximum-minimum distance method to select the third initial cluster center d_3 , such that d_3 satisfies $d_3 = \min \{\max \{s(m_p, d_1), s(m_p, d_2)\}, \dots, s(m_i, m_j, d_{k-1})\}$, removing it from the candidate set. Following this pattern, the next cluster center d_k is selected from the candidate set as $d_k = \min \{\max \{s(m_p, d_1), s(m_p, d_2), \dots, s(m_i, m_j, d_{k-1})\}\}$, resulting in k initial cluster centers. Lines 15-21 obtained the initial clustering results with the obtained k points as initial cluster centers. The distance metric assigns each sample to the cluster center that is closest to it. Lines 22-24 get the final clustering results through iteration.

D. Time Complexity Analysis

The time complexity analysis is as follows. Suppose there are n samples and r features. Generally, n is much larger than r . In addition, let m denote the number of equivalence classes divided by a single attribute, where $m \leq n$. The definition of m will be used in the subsequent time complexity estimation.

In Algorithm 1, the overall structure is an outer loop (Line 1) that iterates r times, once for each feature. Within each iteration of this loop, the time complexity in Line 2 is $O((r-1)n^2)$ as it needs to iterate through each of n samples. The time complexity of Line 3 is $O(n^2)$. Line 4 has a time complexity of $O(1)$, and Lines 5-9 are also $O(m)$. In summary, the main complexity lies in Lines 2-3, which are executed r times due to the outer loop. Consequently, the total time complexity of Lines 1-11 is the result of the outer loop ($O(r)$) multiplied by the complexity of its body (dominated by Line 2, $O(rn^2)$, yielding $O(r \times (O(r-1)n^2 + O(n^2) + O(1) + O(m)))$ or $O(r^2n^2 + rm + r)$. The paper summarizes this as $O(r^2n^2)$, which is the main time cost of Algorithm 1. Therefore, the time complexity of Algorithm 1 is $O(r^2n^2)$.

In Algorithm 2, the cost of Line 1 is $O(r^2n^2)$, based on Algorithm 1, while Lines 2-7 mainly calculate the similarity matrix and have a time complexity of $O(n^2)$. The threshold parameter d is a predefined constant determined by the mean value of the similarity matrix S , which is used to binarize S into the 0-1 matrix S' . As a constant value, d does not affect the asymptotic time complexity. In Lines 8-14, Line 8 classifies samples based on the S' matrix, taking $O(n^2)$. The loop in Lines 9-11 iterates t times to find medoids. The work for each granule X_i (of size N_i) is $O(N_i)$, making the total loop complexity is $O(n)$. The initial center selection in Lines 13-14 takes $O(t^2 + tk)$. Since t and k are typically much smaller than n , this stage is dominated by the $O(n^2)$ granulation step. Lines 16-22 (assignment step) and the update step (implied in Line 23, which takes $O(n^2)$ in the worst case) form one iteration of the clustering. Lines 16-22 represent the main loop, and line 24 represents the loop condition, which repeats for I iterations until convergence. Thus, the total complexity for the iterative clustering (Lines 15-24) is $O(In^2)$. Therefore, the total time complexity of Algorithm 2, which includes Line 1 ($O(r^2n^2)$), Lines 2-7 ($O(n^2)$), and Lines 8-14 ($O(n^2)$), is $(O(r^2n^2 + n^2 + n^2 + In^2) = O((r^2 + I + 2)n^2))$. I represents the number of iterations required for the algorithm to converge, which is typically a constant. In the theoretical worst-case scenario, I could be exponential, but in practice, standard libraries such as scikit-learn set the default upper limit for I to 300 iterations, which indicates that, in practical applications, I is often considered a constant that is much smaller than n . Therefore, it is clear that Line 1 accounts for the majority of the running time of Algorithm 2 in calculating feature weights.

IV. PERFORMANCE ANALYSIS

A. Experimental Setup

Experiments are executed on a computer server equipped with one Intel(R) Core(TM) i7-8550U CPU, 16GB of memory, Windows 10 operating system installed, and implemented using Python 3.10.

Six artificial datasets are used to assess the effectiveness of the proposed IKCS. Detailed information about the 6 artificial datasets is shown in Table III. As shown in Table IV, two two-dimensional artificial simulated datasets M_2 , M_4 are randomly generated, each containing four clusters. The number of samples in datasets M_2 and M_4 is 50 and 400, respectively, resulting in total sample sizes of 200 and 1600. These samples follow a normal distribution, with the mean μ_i and covariance matrix cov_i for the i -th cluster.

TABLE III. ARTIFICIAL DATASETS

DataSets	Samples	features	Clusters
M2	200	2	4
M4	1600	2	4
Compound	399	2	6
Aggregation	788	2	7
Pathbased	300	3	3
Flame	240	2	2

TABLE IV. THE VARIOUS GENERATION PARAMETERS FOR ARTIFICIAL DATASETS

parameters	mean	covariance matrix
M2	$\mu_2 = [2; 2]$	$cov_1 = \begin{bmatrix} 1 & -0.7 \\ -0.7 & 1 \end{bmatrix}$
M4	$\mu_4 = [-2; 2]$	$cov_1 = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$

To evaluate the performance of the IKCS algorithm from multiple dimensions, this study employed the 12 UCI datasets detailed in Table V, which include five high-dimensional and seven low-dimensional datasets. The selection of the five high-dimensional datasets (Ionosphere, Dermatology, Soybean-small, WDBC, Libras Movement) focused on testing the effectiveness of the algorithm in high-dimensional spaces; the seven low-dimensional datasets (Iris, Haberman, Blood, Vertebral, Land-Mines, Maternal-Health, Balance-scale) were chosen to assess the algorithm's benchmark performance in low-dimensional conventional clustering tasks.

TABLE V. UCI DATASETS

DataSets	Samples	features	Clusters
Iris	150	4	3
Haberman	306	3	2
Ionosphere	351	33	2
Blood	748	4	2
Dermatology	366	34	6
vertebral	310	6	2
Land-Mines	338	3	5
Maternal-Health	1013	6	3
Soybean-small	47	35	4
balance-scale	625	4	3
WDBC	569	30	2
Libras Movement	360	90	15

Seven clustering algorithms of fast k -Medoids, PAM, Mini-Batch k -Means, G-Means, DPC, Spectral Clustering, and DBSCAN are applied to compare the performance of the proposed IKCS. Our proposed IKCS algorithm was implemented from scratch using Python 3.10. All baseline algorithms used for performance comparison (fast k -Medoids, PAM, Mini-Batch k -Means, G-Means, DPC, Spectral

Clustering, and DBSCAN) utilized their standard implementations from the `scikit-learn` library. Meanwhile, five evaluation metrics - Accuracy (ACC), Adjusted Rand Index (ARI), Fowlkes-Mallows Index (FMI), Normalized Mutual Information (NMI), and Adjusted Mutual Information (AMI) - are used, with their performance analyzed on both artificial and real datasets. ACC measures classification accuracy, ARI and AMI assess the consistency of clustering results, FMI is based on pairwise relationships between clustering results and true labels, and NMI is based on information theory principles, with higher values indicating better performance. The best experimental results are highlighted in bold.

B. Results on Artificial Datasets

To validate the performance of IKCS on artificial datasets, we conducted experiments on six Artificial datasets listed in Table III, which provides detailed information about these datasets. Table VI presents the clustering results of various algorithms. Meanwhile, Fig. 4 through 7 display the clustering results of each algorithm on the M2, Compound, Aggregation, and Pathbased datasets, respectively, which are consistent with the quantitative data in Table VI. On the relatively simple Fig. 4 (M2 dataset), all tested algorithms (IKCS, fast k -Medoids, PAM, Mini-Batch k -Means, G-Means, and DPC) accurately identify and separate the four distinct clusters. However, in Fig. 5 (Compound dataset), where the distribution structure of sample points is more complex, only IKCS Fig. 5(a) successfully separates all six non-convex clusters; all other comparison algorithms (fast k -Medoids, PAM, Mini-Batch k -Means, G-Means, and DPC) fail to handle the intertwined structures, resulting in incorrect cluster merging. In Fig. 6 (Aggregation dataset), IKCS (Fig. 6(a)) provides the most precise cluster boundaries; while fast k -Medoids, Mini-Batch k -Means, G-Means, and DPC capture the general structure, their boundary definitions are less precise; and PAM (Fig. 6(c)) incorrectly merges two major clusters entirely. Finally, in Fig. 7 (Pathbased dataset), IKCS (Fig. 7(a)), G-Means (Fig. 7(e)), and DPC (Fig. 7(f)) correctly delineate the three path-like clusters; in contrast, fast k -Medoids (Fig. 7(b)) and PAM (Fig. 7(c)) fail to identify any valid structure, and Mini-Batch k -Means (Fig. 7(d)) incorrectly fragments the clusters. These visual results demonstrate the superiority and stability of IKCS in handling diverse and complex data distributions.

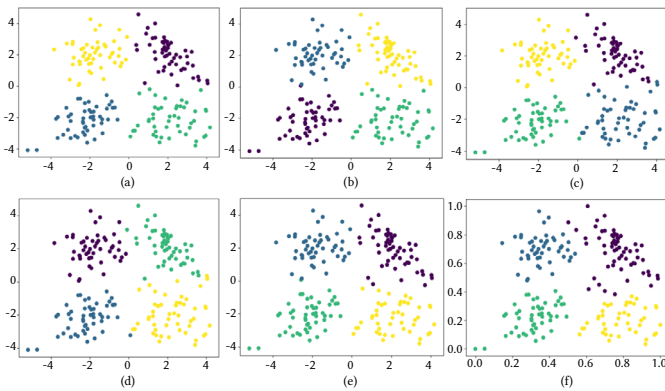


Fig. 4. Clustering results on the M2 dataset: (a) Our IKCS, (b) fast k -Medoids, (c) PAM, (d) Mini-Batch k -Means, (e) G-Means, (f) DPC.

As shown in Table VI, our proposed IKCS demonstrates clustering performance advantages across six artificial datasets. On the M2 and M4 datasets, our algorithm achieves ACC scores of 0.965 and 0.945, respectively, representing improvements of 1.15% to 1.28% over the best-performing comparison algorithms (PAM with 0.9540 on M2 and fast k -Medoids with 0.9437 on M4). Notably, although Mini-Batch k -Means achieves an ACC of 0.9620 on the M2 dataset, its ARI of 0.9011 and AMI of 0.8842 are still 0.75% and 0.04% lower than those of

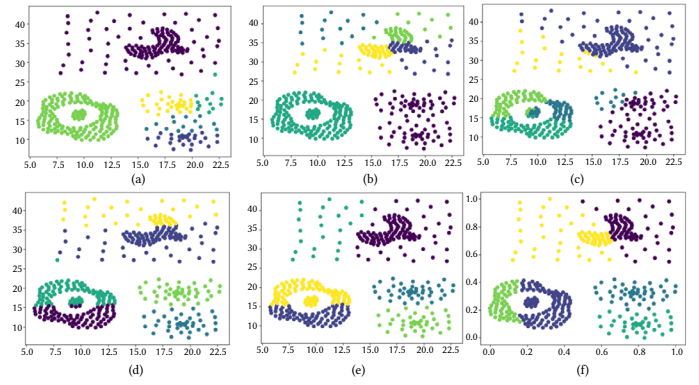


Fig. 5. Clustering results on the Compound dataset: (a) Our IKCS, (b) fast k -Medoids, (c) PAM, (d) Mini-Batch k -Means, (e) G-Means, (f) DPC.

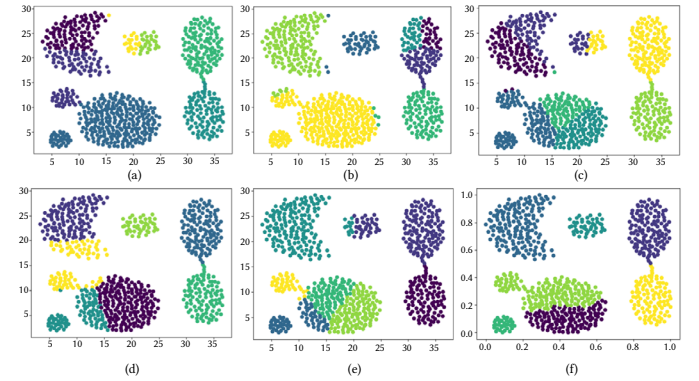


Fig. 6. Clustering results on the Aggregation dataset: (a) Our IKCS, (b) fast k -Medoids, (c) PAM, (d) Mini-Batch k -Means, (e) G-Means, (f) DPC.

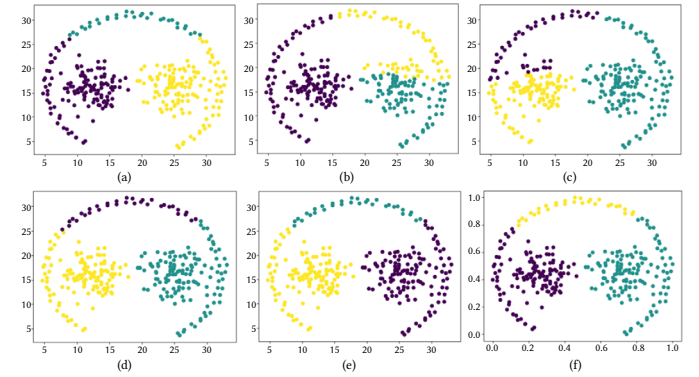


Fig. 7. Clustering results on the Pathbased dataset: (a) Our IKCS, (b) fast k -Medoids, (c) PAM, (d) Mini-Batch k -Means, (e) G-Means, (f) DPC.

our algorithm, indicating that our algorithm not only maintains high accuracy but also more precisely reflects the true cluster structure. On the Compound dataset with complex cluster distributions, our algorithm's ARI of 0.7582 significantly outperforms the second-best algorithm, Fast k -Medoids of 0.7014, by 8.1%, validating its adaptability to non-convex distributions.

On the Aggregation dataset, our algorithm simultaneously achieves the highest values across all five metrics (ACC 0.8764, ARI 0.8169, FMI 0.858, NMI 0.8658, AMI 0.8638), with its NMI showing an average improvement of 4.96% over the comparison algorithms. In contrast, other algorithms exhibit metric imbalances: G-Means achieves the highest AMI of 0.7429 on the Path-based dataset, but its ACC of 0.7525 and FMI of 0.6581 are 1.3% and 0.36% lower than those of our algorithm, respectively. DPC achieves an NMI of 0.4131 on the Flame dataset, close to our algorithm's 0.4394, despite its ACC of 0.7875 lagging

TABLE VI. PERFORMANCE OF 6 CLUSTERING ALGORITHMS ON 6 ARTIFICIAL DATASETS

Dataset	Algorithms	ACC	ARI	FMI	NMI	AMI	Dataset	Algorithms	ACC	ARI	FMI	NMI	AMI
M2	IKCS	0.9650	0.9086	0.9311	0.8865	0.8846	M4	IKCS	0.9458	0.8598	0.8948	0.8235	0.8232
	fast k -Medoids	0.9487	0.8688	0.9016	0.8471	0.8464		fast k -Medoids	0.9437	0.8570	0.8570	0.8221	0.8218
	PAM	0.9540	0.9081	0.9308	0.8841	0.8833		PAM	0.9437	0.8570	0.8927	0.8221	0.8218
	Mini-Batch k -Means	0.9620	0.9011	0.9206	0.8858	0.8842		Mini-Batch k -Means	0.9431	0.8550	0.8912	0.8182	0.8178
	G-Means	0.9600	0.8947	0.9207	0.8782	0.8762		G-Means	0.9381	0.8575	0.8906	0.8128	0.8224
	DPC	0.9500	0.8695	0.9019	0.8760	0.8739		DPC	0.9394	0.8451	0.8813	0.7952	0.7749
Compound	IKCS	0.7744	0.7582	0.8342	0.8093	0.8047	Aggregation	IKCS	0.8764	0.8169	0.8580	0.8658	0.8638
	fast k -Medoids	0.6316	0.7014	0.7787	0.7124	0.7060		fast k -Medoids	0.8185	0.7651	0.8228	0.8243	0.8216
	PAM	0.5238	0.3979	0.5293	0.6271	0.6195		PAM	0.6434	0.5794	0.6643	0.7452	0.7418
	Mini-Batch k -Means	0.6667	0.5472	0.6500	0.7255	0.7198		Mini-Batch k -Means	0.8515	0.7375	0.7940	0.8387	0.8365
	G-Means	0.6591	0.5725	0.6762	0.7687	0.7636		G-Means	0.7525	0.6920	0.7581	0.8449	0.8429
	DPC	0.6667	0.5461	0.6491	0.7365	0.7311		DPC	0.8185	0.7548	0.8072	0.8435	0.8420
Pathbased	IKCS	0.7625	0.7182	0.6604	0.7733	0.7403	Flame	IKCS	0.8542	0.4996	0.7590	0.4394	0.4377
	fast k -Medoids	0.6833	0.3803	0.5988	0.4453	0.4418		fast k -Medoids	0.8100	0.4880	0.7530	0.4320	0.4203
	PAM	0.5967	0.3335	0.5658	0.4127	0.4090		PAM	0.7560	0.2607	0.6712	0.3333	0.3319
	Mini-Batch k -Means	0.7433	0.4624	0.6624	0.5469	0.5439		Mini-Batch k -Means	0.8208	0.4094	0.7148	0.3910	0.3891
	G-Means	0.7525	0.6920	0.6581	0.7449	0.7429		G-Means	0.8333	0.4423	0.7307	0.4120	0.4102
	DPC	0.7333	0.4530	0.6585	0.5390	0.5360		DPC	0.7875	0.3269	0.6786	0.4131	0.4112

TABLE VII. PERFORMANCE OF 8 CLUSTERING ALGORITHMS ON 12 UCI DATASETS

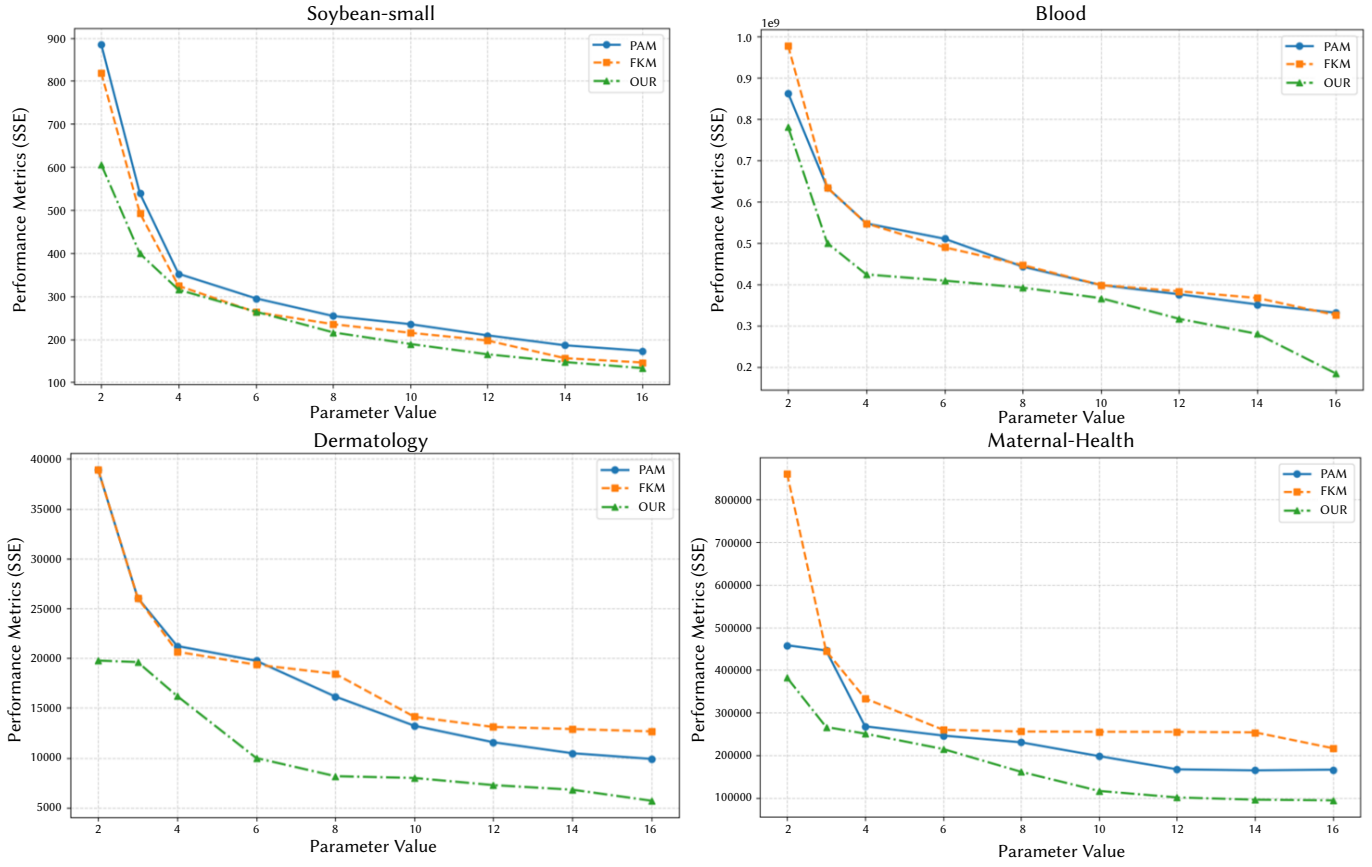
Dataset	Algorithms	ACC	ARI	FMI	NMI	AMI	Dataset	Algorithms	ACC	ARI	FMI	NMI	AMI
Iris	IKCS	0.9067	0.7695	0.8395	0.7857	0.7830	Haberman	IKCS	0.7582	0.1667	0.7376	0.0744	0.0712
	fast k -Medoids	0.9067	0.7583	0.8208	0.7582	0.7551		fast k -Medoids	0.7190	0.0860	0.7076	0.0246	0.0213
	PAM	0.8933	0.7302	0.8208	0.7582	0.7551		PAM	0.7288	0.0951	0.7204	0.0302	0.0268
	Mini-Batch k -Means	0.9067	0.7562	0.8365	0.7571	0.7541		Mini-Batch k -Means	0.5000	0.0037	0.5509	0.0012	0.0014
	G-Means	0.8667	0.6861	0.7969	0.7204	0.7168		G-Means	0.5033	-0.0031	0.5505	0.0010	-0.0016
	DPC	0.5533	0.4531	0.6856	0.6532	0.6483		DPC	0.5817	0.0108	0.5727	0.0024	0.0003
	Spectral Clustering	0.7000	0.5601	0.7538	0.7074	0.7026		Spectral Clustering	0.7386	0.0116	0.7810	0.0145	0.0083
	DBSCAN	0.6667	0.5681	0.7715	0.7337	0.7316		DBSCAN	0.7353	0.0072	0.7774	0.0027	-0.0039
Ionosphere	IKCS	0.7094	0.1729	0.6033	0.1320	0.1301	Blood	IKCS	0.6805	0.0725	0.7013	0.0411	0.0405
	fast k -Medoids	0.6923	0.1456	0.6002	0.0529	0.0503		fast k -Medoids	0.6230	0.0505	0.4235	0.0089	0.0070
	PAM	0.6783	0.1728	0.6028	0.1230	0.1121		PAM	0.6459	0.0653	0.5133	0.0113	0.0110
	Mini-Batch k -Means	0.6866	0.1343	0.5920	0.0854	0.0835		Mini-Batch k -Means	0.6687	0.0583	0.6901	0.0282	0.0262
	G-Means	0.6866	0.1050	0.6678	0.0584	0.0559		G-Means	0.6553	0.0212	0.6811	0.0043	0.0023
	DPC	0.5185	-0.0323	0.5984	0.0860	0.0837		DPC	0.5668	0.0144	0.5729	0.0397	0.0387
	Spectral Clustering	0.6439	0.0045	0.7331	0.0087	0.0032		Spectral Clustering	0.6043	0.0185	0.5964	0.0033	0.0022
	DBSCAN	0.6949	0.1501	0.5284	0.1131	0.0327		DBSCAN	0.5094	-0.0036	0.5451	0.0026	-0.0002
Dermatology	IKCS	0.3314	0.0570	0.2411	0.1392	0.1204	Vertebral	IKCS	0.7323	0.2130	0.7609	0.2507	0.2488
	fast k -Medoids	0.2569	0.0192	0.2260	0.1108	0.0907		fast k -Medoids	0.6742	-0.0034	0.7459	0.0039	-0.0024
	PAM	0.2762	0.0278	0.2336	0.1148	0.0942		PAM	0.6710	0.1051	0.5982	0.2353	0.2334
	Mini-Batch k -Means	0.3066	0.0392	0.2269	0.1130	0.0936		Mini-Batch k -Means	0.6645	0.0955	0.5956	0.2493	0.2474
	G-Means	0.3122	0.0546	0.2397	0.1349	0.1161		G-Means	0.6677	0.1003	0.5969	0.2323	0.2304
	DPC	0.3054	0.0521	0.2261	0.0702	0.0951		DPC	0.6516	0.1029	0.5914	0.1274	0.1250
	Spectral Clustering	0.3011	0.0452	0.2291	0.1318	0.1130		Spectral Clustering	0.5419	-0.0580	0.6223	0.0671	0.0642
	DBSCAN	0.3122	0.0021	0.2482	0.0430	0.0158		DBSCAN	0.6742	-0.0034	0.7459	0.0039	-0.0024
Soybean-small	IKCS	0.8085	0.5970	0.7456	0.7891	0.7740	Balance-scale	IKCS	0.5664	0.1998	0.5051	0.1741	0.1714
	fast k -Medoids	0.7234	0.5046	0.6812	0.7024	0.4954		fast k -Medoids	0.5296	0.1333	0.4655	0.1015	0.0986
	PAM	0.5106	0.3041	0.4799	0.4731	0.4258		PAM	0.4192	0.0346	0.4049	0.0336	0.0305
	Mini-Batch k -Means	0.7234	0.5452	0.6568	0.7158	0.6909		Mini-Batch k -Means	0.5392	0.1742	0.4941	0.1523	0.1496
	G-Means	0.4894	0.1641	0.5153	0.2668	0.2379		G-Means	0.5296	0.1183	0.4576	0.0983	0.0954
	DPC	0.6509	0.4821	0.6621	0.6521	0.6639		DPC	0.5248	0.0412	0.4736	0.0776	0.0742
	Spectral Clustering	0.6809	0.5924	0.7327	0.7879	0.7659		Spectral Clustering	0.5376	0.1334	0.4650	0.1091	0.1062
	DBSCAN	0.3191	-0.0194	0.4650	0.0587	0.0109		DBSCAN	0.3191	-0.0194	0.4650	0.0587	0.0109

by 7.67%. This multi-dimensional advantage confirms the superior consistency of our algorithm across various evaluation metrics.

In noise-sensitive scenarios, this algorithm demonstrates stronger robustness. Traditional algorithms expose their limitations in specific scenarios: In the Compound dataset, the ACC of PAM drops sharply to 0.5238 (a 45.1% decrease compared to M2), while this algorithm maintains a stable performance with an ACC > 0.7625 across six types of

data distributions with significant differences. In the Flame dataset, the ARI of this algorithm (0.4996) is 2.38% higher than that of the suboptimal fast k -Medoids (0.488), and the FMI index is 0.79% ahead, demonstrating its advantage in identifying density-heterogeneous clusters.

Overall, our proposed method consistently achieves better performance across various datasets, making it the most reliable and versatile option.

Fig. 8. Performance of k -choice on various datasets.

C. Experiment on UCI Datasets

To verify the performance of IKCS on real-world datasets, we conducted experiments on twelve UCI datasets, as shown in Table V. The table provides detailed information about these datasets, while Table VII presents the clustering results of various algorithms. As shown in Table VII, the IKCS algorithm demonstrates superior clustering performance on 12 UCI datasets.

In the Iris dataset, IKCS outperforms the comparison algorithm in five indicators, including $ACC=0.9067$ and $ARI=0.7695$. Its NMI value (0.7857) is 2.75% higher than the second-best fast k -Medoids algorithm. It is worth noting that in the high-dimensional soybean small dataset (dimension=35), the AMI metric of IKCS reaches 0.7740, which is 0.81% higher than that of the spectral clustering algorithm, verifying the strong adaptability of the algorithm to the feature space. Traditional algorithms show obvious limitations in complex data scenarios. In the dermatology dataset (number of categories=6), although the absolute performance of all algorithms is limited, IKCS still maintains a comparative advantage, $ACC=0.3314$, which is 1.92% higher than the best comparison algorithm, G-Means. DPC algorithm reveals the imbalance of measurement in this dataset, and the deviation between AMI (0.0951) and ACC (0.3054) is 21.03%. On the noise-sensitive Haberman data set, the FMI value of IKCS (0.7376) is 1.72% higher than that of PAM, showing stronger robustness. IKCS exhibits stability across various data types. In the Land Mines dataset, the ARI measurement is 0.0599, which is slightly lower than the G-Means. In particular, in the Vertebral dataset, IKCS outperforms in five indicators, demonstrating the high consistency of the algorithm across multiple measurement dimensions. On the contrary, the G-Means algorithm displays a negative ARI value (-0.0031) in the Haberman dataset, and IKCS effectively avoids this exception by introducing a weighting mechanism.

The Mini-Batch k -Means algorithm performs well on some datasets, such as the ACC of the Iris dataset, which is 0.9067. Still, it performs poorly on other datasets, such as the ACC of the Dermatology dataset, which is only 0.3066. This shows that the algorithm is highly dependent on data sets. The G-Means algorithm typically performs well, especially in ARI and AMI indicators. For example, in the Dermatology dataset, the ARI of G-Means is only 0.0546. The DPC algorithm performs poorly on most datasets, particularly with respect to ARI and AMI indicators. For example, in the Dermatology dataset, the ARI of the DPC is -0.0323, showing a poor clustering effect. The Spectral Clustering algorithm performs well on some datasets, notably the ACC in the Iris dataset, which is 0.7000. Still, it performs poorly on other datasets, such as the ACC in the Dermatology dataset, which is only 0.3011. The overall performance of the DBSCAN algorithm is relatively general.

Overall, the IKCS algorithm demonstrates greater stability and performance across multiple datasets, while traditional algorithms such as DPC and G-Means yield more average results, highlighting the algorithm's effectiveness.

D. The Effect of SSE at Different k values

To test the sensitivity of IKCS to the key parameter k , this section uses the Sum of Squared Errors (SSE) as a metric. The Sum of Squared Errors (SSE) is a key metric for evaluating intra-cluster compactness, where lower SSE values indicate superior clustering quality. Fig. 8 is designed to assess the performance stability of IKCS across different dimensional scenarios, comparing its results on two high-dimensional datasets (Soybean-small, 35 features; Dermatology, 34 features) and two low-dimensional datasets (Blood, 4 features; Maternal-Health, 8 features). As depicted in Fig. 8, across all four datasets, the SSE curve of IKCS remains stable over the entire tested range of k , without its

TABLE VIII. COMPARISON OF CLUSTERING ALGORITHMS RUNNING TIME (SECONDS)

Dataset	IKCS	FKM	PAM	MiniBatch	G-Means	DPC	Spectral Cl	DBSCAN
Iris	0.182486	0.078644	0.081204	0.316647	0.078652	0.619395	0.145073	0.004002
Haberman	0.651097	0.104465	0.069878	0.346992	0.072482	1.226336	0.179395	0.005996
Ionosphere	2.979099	0.190845	0.090910	0.370084	0.092433	1.344881	0.237368	0.007833
Blood	4.129927	0.346581	0.084169	0.350662	0.101401	3.232304	0.387127	0.012733
Dermatology	4.047163	0.120343	0.123642	0.351267	0.098178	1.413009	0.259473	0.008077
Vertebral	1.139654	0.387302	0.369440	0.552829	0.114839	1.189338	0.452543	0.008019
Land-Mine	0.940599	0.114076	0.094792	0.352368	0.100983	1.527057	0.188698	0.005939
Maternal-Health	8.172306	0.450520	0.116328	0.385469	0.145879	4.944714	0.594744	0.025618
Soybean-small	0.072797	0.073127	0.061086	0.337883	0.081384	0.254308	0.131675	0.002998
Balance-scale	3.383729	0.247438	0.081283	0.416138	0.177840	2.575240	0.340651	0.007610
WDBC	9.232958	0.240365	0.148639	0.326162	0.103066	2.371847	0.295971	0.012026
Libras Movement	9.962579	0.373323	0.357271	0.722531	0.162811	1.482681	0.472559	0.023998

performance diminishing significantly as k changes, indicating that IKCS is not highly sensitive to the parameter k , does not rely on a specific value, and maintains its effectiveness across a wide range, thus demonstrating high robustness. Concurrently, the SSE curve of IKCS remains the lowest for all tested values of k , which shows its superior clustering quality. Furthermore, the clustering quality advantage of IKCS is more pronounced on high-dimensional data. The margin of its clustering quality advantage on the two high-dimensional datasets (Soybean-small and Dermatology) is significantly larger than the margin of its advantage on the two low-dimensional datasets (Blood and Maternal-Health), indicating that the feature weighting mechanism of IKCS provides a performance improvement in high-dimensional scenarios by identifying key features and reducing the impact of noisy ones.

E. Running Time

To ensure reliable and fair timing comparisons and to mitigate the impact of stochastic factors, such as cache memory, each execution time reported in Table VIII is the arithmetic mean of 20 independent runs, excluding any dataset loading time. As shown in Table VIII, the proposed IKCS needs to select a good initial cluster center in advance to avoid the limitation of random selection, so it needs to form an initial particle set, and then select the initial center with the most significant density and large distance separation according to the maximum and minimum distance method. Therefore, the algorithm takes a long time to process datasets with large scales and dimensions, such as the Maternal Health and WDBC, among others. However, the running time of IKCS on some small datasets is not the longest. For example, on Iris, Haberman, Vertebral, and other datasets, the DPC algorithm takes longer. This is due to the DPC needing to calculate the local density and distance of each data point on small datasets, resulting in high calculation complexity and longer running times. Other algorithms have a shorter running time, but their clustering results are not as good as those of IKCS. In short, the selection of the clustering algorithm must be balanced according to the dataset's size and dimension.

V. CONCLUDING REMARKS AND FUTURE WORK

In this work, we define the granularity rough entropy using the relevant theories of knowledge granularity and rough sets, and use it to calculate the importance of features. Next, we introduce feature weight coefficients to define the similarity function between samples and integrate these feature weights into the similarity calculation of the fast k -Medoids clustering algorithm. In the proposed IKCS, we addressed the common problem of initial cluster centers being very close or belonging

to the same category in the fast k -Medoids algorithm. Experimental results on both artificial and UCI datasets support the superiority of our proposed IKCS, which addresses some of the limitations of the fast k -Medoids and significantly improves the stability and accuracy of the clustering results. These findings demonstrate the high feasibility and effectiveness of the proposed method.

The proposed approach offers dual advantages. First, it maximizes feature distinguishability through weighted allocation based on importance. Second, it strengthens clustering robustness by utilizing a coarse-grained entropy technique to optimize the selection of initial cluster centers. Nevertheless, the method also has limitations that include high computational complexity—particularly in data analysis and weight calculation—sensitivity to thresholds, and inadequate management of high-dimensional sparsity. As future work, we aim to focus on incorporating Locally Sensitive Hashing (LSH) and GPU acceleration or designing dynamic threshold mechanisms to reduce parameter dependency issues.

FUNDING

This work is supported in part by the National Natural Science Foundation of China under grants U2468205 and 62472168, the Natural Science Foundation of Hunan Province under grant 2025JJ50399, the Hunan Provincial Key Research and Development Program under Grant 2024AQ2028, the Scientific Research Project of the Education Department of Hunan Province under Grant 25A0370, the Social Science Foundation of Hunan Province under Grant 25YBA142 and the Teaching Reform Project of Hunan Provincial Undergraduate Universities under Grant 202502000842.

CREDiT AUTHORSHIP CONTRIBUTION STATEMENT

L. Chen: Conceptualization, methodology, data curation, writing—original draft preparation, visualization, supervision, funding acquisition.

L. Li: Methodology, software, validation, formal analysis, investigation, writing—original draft preparation, visualization.

X. Ye: Methodology, software, validation, investigation, writing—original draft preparation.

Y. Chen: Formal analysis, resources, writing—review and editing, funding acquisition.

W. Liang: Resources, writing—review and editing.

K. Li: Conceptualization, resources, writing—review and editing, supervision, funding acquisition.

DATA STATEMENT

The public datasets used in this study are openly available from their original sources.

DECLARATION OF CONFLICTS OF INTEREST

On behalf of all authors, the corresponding author states that there is no conflict of interest.

REFERENCES

- [1] L. Peng, J. Gao, Z. Yang, X. Ai, W. Liang, "Chemistry-structure dual-perception large language models: Advancing molecular property prediction for precise disease treatment," *IEEE Journal of Biomedical and Health Informatics*, 2026.
- [2] J. Zhang, P. Gou, M. Kong, K. Fang, J. Xiao, Q. Zhang, X. Xin, J. Yu, "Pam-8 im/dd transmission based on modified lookup table nonlinear predistortion," *IEEE Photonics Journal*, vol. 10, no. 3, pp. 1–9, 2018.
- [3] P. Gou, L. Zhao, K. Wang, W. Zhou, J. Yu, "Nonlinear look-up table predistortion and chromatic dispersion precompensation for im/dd pam-4 transmission," *IEEE Photonics Journal*, vol. 9, no. 5, pp. 1–7, 2017.
- [4] N. Sureja, B. Chawda, A. Vasant, "An improved k-medoids clustering approach based on the crow search algorithm," *Journal of Computational Mathematics and Data Science*, vol. 3, p. 100034, 2022.
- [5] Z. Wu, L. Jin, J. Zhao, L. Jing, L. Chen, "Research on segmenting e-commerce customer through an improved k-medoids clustering algorithm," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 9930613, 2022.
- [6] C. Lucasius, A. Dane, G. Kateman, "On k-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasibility and comparison," *Analytica Chimica Acta*, vol. 282, no. 3, pp. 647–669, 1993.
- [7] J. Xie, R. Girshick, A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478–487, PMLR.
- [8] Y. Tang, Z. Pan, W. Pedrycz, F. Ren, X. Song, "Based kernel fuzzy clustering with weight information granules," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 2, pp. 342–356, 2022.
- [9] L. Sun, Q. Zhang, W. Ding, T. Wang, J. Xu, "Fcdfs: fuzzy granular ball clustering-based partial multilabel feature selection with fuzzy mutual information," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 9, no. 1, pp. 590–606, 2024.
- [10] R. Ng, J. Han, "Clarans: A method for clustering objects for spatial data mining," *IEEE transactions on knowledge and data engineering*, vol. 14, no. 5, pp. 1003–1016, 2002.
- [11] T. Deng, G. Yang, Y. Huang, M. Yang, H. Fujita, "Adaptive multi-granularity sparse subspace clustering," *Information Sciences*, vol. 642, p. 119143, 2023.
- [12] Z. Zhao, S. Guo, Q. Xu, T. Ban, "G-means: a clustering algorithm for intrusion detection," in *International Conference on Neural Information Processing*, 2008, pp. 563–570, Springer.
- [13] J. Hou, A. Zhang, N. Qi, "Density peak clustering based on relative density relationship," *Pattern Recognition*, vol. 108, p. 107554, 2020.
- [14] J. Liu, J. Han, "Spectral clustering," in *Data clustering*, Chapman and Hall/CRC, 2018, pp. 177–200.
- [15] K. Khan, S. Rehman, K. Aziz, S. Fong, S. Sarasvady, "DbSCAN: Past, present and future," in *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, 2014, pp. 232–238, IEEE.
- [16] R. Behzadidoost, F. Mahan, H. Izadkhah, "Granular computing-based deep learning for text classification," *Information Sciences*, vol. 652, p. 119746, 2024.
- [17] Y. Yao, "Perspectives of granular computing," in *2005 IEEE international conference on granular computing*, vol. 1, 2005, pp. 85–90, IEEE.
- [18] J. Feng, J. Chen, "A hierarchical clustering algorithm based on granular-ball and shared natural nearest neighbour," in *2025 6th International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, 2025, pp. 426–430, IEEE.
- [19] X. Bao, D. Wang, T. Wu, Z. Yu, "Probabilistic time series forecasting based on the principle of justifiable granularity," in *2024 9th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 2024, pp. 517–522, IEEE.
- [20] L. Chen, P. Chen, A. Hu, W. Liang, "A proposed weighted k-medoids clustering method and its application on improving higher education quality," in *2024 5th International Conference on Information Science and Education (ICISE-IE)*, 2024, pp. 406–413, IEEE.
- [21] S. Sun, L. Chen, B. Mei, T. Li, X. Ye, M. Shi, "A weighted k-medoids clustering algorithm based on granular computing," in *2023 IEEE 10th International Conference on Cyber Security and Cloud Computing (CSCloud)/2023 IEEE 9th International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, 2023, pp. 138–143, IEEE.
- [22] L. Chen, T. Li, Y. Chen, X. Chen, M. Wozniak, N. Xiong, W. Liang, "Design and analysis of quantum machine learning: a survey," *Connection Science*, vol. 36, no. 1, p. 2312121, 2024.
- [23] L. Chen, F. Shen, Y. Tang, X. Wang, J. Wang, "Algebraic structure based clustering method from granular computing perspective," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 31, no. 01, pp. 121–140, 2023.
- [24] L. Chen, L. Zhao, Z. Xiao, Y. Liu, J. Wang, "A granular computing based classification method from algebraic granule structure," *IEEE Access*, vol. 9, pp. 68118–68126, 2021.
- [25] L. Chen, J. Wang, W. Wang, L. Li, "A new granular computing model based on algebraic structure," *Chinese Journal of Electronics*, vol. 28, no. 1, pp. 136–142, 2019.
- [26] L. Chen, J. Wang, L. Li, "The models of granular system and algebraic quotient space in granular computing," *Chinese Journal of Electronics*, vol. 25, no. 6, pp. 1109–1113, 2016.
- [27] J. Cai, W. Liang, X. Li, K. Li, Z. Gui, M. Khan, "Gtxchain: A secure iot smart blockchain architecture based on graph neural network," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21502–21514, 2023.
- [28] X. Wang, L. Zhou, K. Li, S. Zheng, H. Fan, "Iatraj: multi-modal trajectory prediction through contextual information spatio-temporal interaction and awareness," 2025.
- [29] S. Zhou, K. Li, Y. Chen, C. Yang, W. Liang, A. Zomaya, "Trustbcl: Mitigating data bias in iot through blockchain-enabled federated learning," *IEEE Internet of Things Journal*, vol. 11, no. 15, pp. 25648–25662, 2024.
- [30] S. Zhou, Y. Chen, K. Li, W. Liang, W. Meng, "Trusthfl: An efficient aggregation method for trustworthy hierarchical federated learning," *IEEE Internet of Things Journal*, 2026.
- [31] W. Liang, S. Xie, K. Li, X. Li, X. Kui, A. Zomaya, "Mc-dsc: A dynamic secure resource configuration scheme based on medical consortium blockchain," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 3525–3538, 2024.
- [32] J. Long, W. Liang, K. Li, Y. Wei, M. Marino, "A regularized cross-layer ladder network for intrusion detection in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1747–1755, 2022.
- [33] C. Diao, D. Zhang, W. Liang, K. Li, Y. Hong, J. Gaudiot, "A novel spatial-temporal multi-scale alignment graph neural network security model for vehicles prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 904–914, 2022.
- [34] W. Liang, Y. Liu, C. Yang, S. Xie, K. Li, W. Susilo, "On identity, transaction, and smart contract privacy on permissioned and permissionless blockchain: A comprehensive survey," *ACM Computing Surveys*, vol. 56, no. 12, pp. 1–35, 2024.
- [35] H. Zhang, Y. Chen, K. Li, Y. Li, S. Zhou, W. Liang, A. Poniszewska-Maranda, "Robust federated learning with contrastive learning and meta-learning," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 9, no. 6, pp. 38–51, 2026.
- [36] C. Liang, Z. Diao, X. Wang, Y. Huo, K. Li, D. He, W. Liang, "Fedahpip: Federated learning with adaptive hot parameter identification and personalized anchoring for multi-agent collaboration," *Journal of Industrial Information Integration*, p. 101087, 2026.
- [37] Y. Li, W. Liang, K. Xie, D. Zhang, S. Xie, K. Li, "Lightnestle: quick and accurate neural sequential tensor completion via meta learning," in *IEEE INFOCOM 2023-IEEE conference on computer communications*, 2023, pp. 1–10, IEEE.
- [38] J. Liao, L. Guo, L. Jiang, C. Yu, W. Liang, K. Li, F. Pop, "A machine

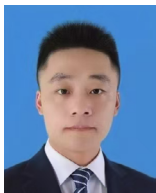
learning-based feature extraction method for image classification using resnet architecture,” *Digital Signal Processing*, vol. 160, p. 105036, 2025.

- [39] A. Rodriguez, A. Laio, “Clustering by fast search and find of density peaks,” *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.
- [40] M. K. Dhar, S. N. Hasan, T. R. Otushi, M. Khan, “Entropy-based feature selection for data clustering using k-means and k-medoids algorithms,” in *2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, 2020, pp. 36–40, IEEE.
- [41] N. Amalia, *et al.*, “Improving k-medoids clustering with z-score initialization: A novel approach to reducing iterations and enhancing performance,” in *2025 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, 2025, pp. 1–6, IEEE.
- [42] X. Jiang, D. Cheng, J. Huang, S. Zhang, “Fast minimum spanning tree clustering algorithm based on granular-ball computing,” in *International Joint Conference on Rough Sets*, 2025, pp. 405–419, Springer.
- [43] Z. Pawlak, *Rough sets: Theoretical aspects of reasoning about data*, vol. 9. Springer Science & Business Media, 2012.
- [44] H. Ji, K. Xie, J. Wen, Q. Zhang, G. Xie, W. Liang, “Finemon: An innovative adaptive network telemetry scheme for fine-grained, multi-metric data monitoring with dynamic frequency adjustment and enhanced data recovery,” *Proceedings of the ACM on Management of Data*, vol. 2, no. 1, pp. 1–26, 2024.
- [45] L. Peng, C. Yang, J. Yang, Y. Tu, Q. Yu, Z. Li, M. Chen, W. Liang, “Drug repositioning via multi-view representation learning with heterogeneous graph neural network,” *IEEE Journal of Biomedical and Health Informatics*, vol. 29, no. 3, pp. 1668–1679, 2024.
- [46] Z. Pawlak, “Rough sets and intelligent data analysis,” *Information sciences*, vol. 147, no. 1–4, pp. 1–12, 2002.



Linshu Chen

Linshu Chen received a B.S. degree in computer science from the Hunan University of Science and Technology in 2004, and the M.S. and Ph.D. degrees in computer science from Central South University in 2007 and 2017. From 2019 to 2020, he was a visiting scholar at Florida Atlantic University. His research interests include machine learning, intelligent information processing, information fusion, and GrC.



Liang Li

Liang Li earned his B.S. in Computer Science from Xiangtan Institute of Technology in 2024 and is now pursuing an M.S. in Computer Science at Hunan University of Science and Technology. His work centers on cutting-edge machine learning and GrC, where he is driven to explore innovative methods and impactful applications.



Xueqi Ye

Xueqi Ye received the B.S. degree in computer science from Xiaoxiang College of Hunan University of Science and Technology in 2023 and is currently pursuing an M.S. degree in computer science at Hunan University of Science and Technology, with research interests that include machine learning, clustering, and GrC.



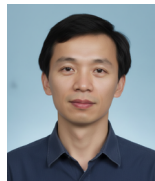
Yuxiang Chen

Yuxiang Chen received a PhD degree in computer application from Hunan University, Changsha, China, in 2021. He is currently an assistant professor at Hunan University of Science and Technology. His research interests include network monitoring, network security, big data, and Machine Learning.



Wei Liang

Wei Liang received a Ph.D. degree in computer science and technology from Hunan University in 2013, and was a postdoctoral scholar with Lehigh University during 2014–2016. Dr. Liang is currently a Professor at the School of Computer Science and Engineering, Hunan University of Science and Technology. His research interests include intelligent transportation, security of IoV, blockchain, hardware IP protection, and security management in wireless sensor networks.



Kuanching Li

Kuanching Li is a professor at the Hunan University of Science and Technology. Dr. Li has co-authored over 150 conference and journal papers, holds several patents, and serves as an associate and guest editor for various scientific journals. He has also held chair positions at several prestigious international conferences. His research interests include cloud and edge computing, Big Data, and blockchain technologies.