KMHC: A Parallel Hybrid Clustering Algorithm Integrating K-Means and Hierarchical Agglomerative Clustering with Voting-Based Consensus

Aida Chefrour[1,2], Soufiane Khedairia[3]
[1]Computer Science Department, Science and Technology Faculty, Souk Ahras University, 41000, Algeria
[2]LISCO Laboratory, Badji Mokhtar University, Annaba, B.P-12, 23000, Algeria
[3]LiM Laboratory, Department of Computer Science, Faculty of Science and Technology, University of Souk Ahras, Souk Ahras, 41000, Algeria
E-mail: aida.chefrour@univ-soukahras.dz, soufiane.khedairia@univ-soukahras.dz

Keywords: Hybrid clustering k-means, hierarchical agglomerative clustering, overlap-based label alignment, voting consensus, KMHC

Received: September 1, 2025

This paper proposes KMHC, a parallel hybrid clustering algorithm that combines K-means and Hierarchical Agglomerative Clustering (HAC) through an overlap-based label alignment and deterministic voting consensus mechanism. The objective of KMHC is to exploit the complementarity between centroid-based compactness and hierarchical structural information in order to improve clustering consistency and reduce assignment errors. Unlike conventional ensemble clustering methods that rely on repeated resampling, large pools of base partitions, or co-association matrices, KMHC uses a lightweight two-clusterer architecture with explicit cluster correspondence estimation. The proposed method is formally defined through an overlap matrix, a cluster mapping function, and a deterministic tie-breaking rule. Its computational complexity is also analyzed, showing that the main bottleneck is the HAC component, while the alignment and voting stages introduce limited additional cost. KMHC is evaluated on benchmark and synthetic datasets with different dimensionalities, noise levels, and cluster distributions. Its performance is compared with K-means, HAC, DBSCAN, Spectral Clustering, and Gaussian Mixture Models using precision, normalized mutual information, adjusted Rand index, error rate, runtime, and stability over multiple independent runs. Statistical significance tests and an ablation study are also conducted to assess the contribution of K-means, HAC, label alignment, and voting. The results show that KMHC can improve clustering performance when the two base clusterers provide complementary partitions, particularly in datasets where both local compactness and hierarchical structure are informative. However, the method remains limited by the quadratic cost of HAC and by its current batch-mode design. Future work will investigate approximate hierarchical clustering, incremental extensions, and streaming-data scenarios.

Povzetek: Članek predstavlja KMHC, hibridni algoritem za gručenje, ki združuje K-means in HAC z usklajevanjem oznak ter deterističnim glasovanjem. Rezultati kažejo izboljšano gručenje pri komplementarnih particijah, vendar metodo omejujeta zahtevnost HAC in paketna obdelava.

1 Introduction

Machine learning has experienced a surge in popularity in recent years, demonstrating its significant potential in various applications [1]. These include data mining, natural language processing, image recognition, and expert systems. Machine learning is poised to become the cornerstone of our future society by offering solutions in these and other domains. There are two primary approaches within machine learning: supervised and unsupervised learning. Supervised learning uses a predefined set of training examples to enable the program to conclude when presented with new data. In contrast, unsupervised learning, or clustering, involves programs that identify patterns and relationships within data without prior training examples.

Clustering, a complex challenge within data mining and machine learning, plays a crucial role in data analysis [2]. By uncovering the relationships between and within data objects and entities, clustering proves invaluable in various analytical problems, including customer segmentation in marketing, image segmentation in computer vision, and document clustering in information retrieval. These applications involve issues related to set partitioning, cluster positioning, distance metrics, and types of partitioning methods [3]. However, clustering also presents challenges, such as determining the optimal number of clusters and handling high-dimensional data. Recent research has addressed these challenges with advancements in ensemble clustering and density-based methods. In our research, we focus on developing a novel clustering algorithm that effectively leads to improved accuracy in data classification tasks.

Our exploration of machine learning focuses on two prominent algorithms: k-means clustering [4], a partitioning method, and ascending hierarchical classification [5]. Traditionally, pattern recognition systems relied on testing and evaluating individual classifiers, ultimately selecting the one with the highest performance. However, it became evident that combining classifiers could leverage their complementary strengths, leading to the emergence of classifier combination techniques. These techniques offer a powerful means to achieve superior performance without increasing the complexity of existing classification methods, making them particularly suitable for applications demanding high accuracy. Various combination methods exist [6], including simple voting and weighted averaging, each with its own advantages and limitations.

Clustering algorithms operate on the principle that similar data points should be grouped together, while dissimilar points should reside in separate clusters. However, many clustering methods, such as k-means and ascendant hierarchical classification, often lead to varying results on the same dataset. This lack of a universally optimal method poses a challenge, particularly when users cannot specify the appropriate clustering criteria before hand [7].

The main contributions of this work are summarized as follows:

- A lightweight hybrid clustering architecture (KMHC) that combines K-means and Hierarchical Agglomerative Clustering (HAC) in a parallel, non-iterative pipeline, enabling a balance between speed and precision for large-scale datasets. Unlike traditional ensemble clustering approaches that rely on bagging or resampling, KMHC adopts a direct consensus strategy to fuse complementary structures extracted from different clustering perspectives.
- An adaptive parallel voting mechanism that aligns cluster labels using an overlap-based mapping heuristic. This voting procedure improves cluster consistency by reconciling label permutations between different base clusterers, especially for high-dimensional data where label correspondence is ambiguous.
- A thorough empirical validation across three public benchmark datasets (Gisette, Letter Recognition, Optdigits), demonstrating that KMHC can outperform base classifiers by up to 30% in precision. Additionally, we perform a quantitative comparison with state-of-the-art clustering methods, including DBSCAN, Spectral Clustering, and Gaussian Mixture Models, to show the competitiveness of KMHC in terms of clustering quality and error rate.

It is important to distinguish KMHC from conventional ensemble and consensus clustering frameworks. Many ensemble clustering methods generate diversity by repeatedly applying the same clustering algorithm with different initializations, parameters, or resampled subsets of the data. In contrast, KMHC achieves diversity through algorithmic heterogeneity by combining two different clustering principles: the centroid-based compactness of K-means and the hierarchical structural information of HAC. Moreover, KMHC does not construct a large ensemble of partitions or a full co-association matrix. Instead, it uses a lightweight overlap-based label alignment followed by a deterministic voting rule. Therefore, the novelty of KMHC lies not only in combining two clustering algorithms, but also in its simple, reproducible, and computationally limited consensus strategy.

The main objective of this study is to investigate whether a lightweight parallel hybrid clustering framework can improve clustering performance by combining the complementary properties of K-means and Hierarchical Agglomerative Clustering (HAC). More specifically, the study addresses the following research questions:

- **RQ1:** Can a parallel hybrid clustering approach combining K-means and HAC improve clustering quality compared with the two individual clustering algorithms?
- **RQ2:** Does overlap-based label alignment reduce inconsistencies caused by label permutation between heterogeneous clustering outputs?
- **RQ3:** How does KMHC compare with representative clustering baselines in terms of precision, normalized mutual information, adjusted Rand index, error rate, runtime, and stability?
- **RQ4:** What are the computational limitations of KMHC under high-dimensional, noisy, imbalanced, or large-scale data conditions?

Based on these research questions, the study formulates the following hypotheses:

- **H1:** Combining K-means and HAC can improve clustering quality when the two algorithms capture complementary structures in the data.
- **H2:** Explicit overlap-based label alignment improves the reliability of the consensus mechanism by addressing the label permutation problem.
- **H3:** The voting-based KMHC framework can improve stability across multiple runs compared with individual clustering algorithms, while maintaining moderate computational cost for small- and medium-scale datasets.

The intended outcomes of the study are therefore to evaluate KMHC in terms of clustering performance, statistical stability, computational complexity, and component-level contribution through ablation analysis.

The rest of the paper is organized as follows: In the next section, we briefly review the literature on hybrid clustering algorithm overviews. We describe the proposed

KMHC clustering algorithm and introduce some representative methods in Section 3. Section 4 describes the experiment we conducted and the results obtained by our algorithm. Finally, we draw some conclusions and show ongoing research aspects in Section 5.

2 Related works

Clustering algorithms play a vital role in data analysis, enabling the discovery of natural groupings within data and providing valuable information for various applications. Although there are numerous clustering algorithms, each with its own strengths and limitations, combining multiple algorithms has emerged as a promising approach to overcome individual weaknesses and achieve improved clustering performance. This related work section delves into existing research on clustering algorithm combinations, exploring different approaches such as ensemble clustering, hybrid clustering, and consensus clustering.

The research of [8] proposed a novel hybrid clustering algorithm that enhances density peak clustering (DPC) by addressing its limitations. The algorithm incorporates several key improvements: a density calculation based on a k-closet algorithm for increased accuracy, an advanced decision graph and selection strategy for reliable cluster center identification, k-means clustering for refined cluster assignments; and a noise identification mechanism for improved robustness. Experiments show that these improvements work, as they lead to better accuracy, greater robustness, and more consistent performance across different datasets compared to traditional DPC and other clustering methods. In general, this hybrid approach offers a promising solution to achieve improved clustering results by combining the strengths of both DPC and k-means while overcoming their individual weaknesses. This makes it a valuable tool for researchers and practitioners working with complex data analysis tasks that require accurate and robust clustering solutions.

The study of [9] presented a hybrid clustering approach that combines genetic algorithms (GA) with k-means and k-medoids algorithms to improve clustering performance. GA is used to optimize the initial cluster centers for both k-means and k-medoids, addressing a key weakness of these algorithms: sensitivity to initial center selection. You can think of cluster centers as chromosomes in a GA population. In this method, a fitness function based on cluster quality metrics, such as the sum of squared errors, governs the evolutionary process in this method. By evolving a population of potential solutions, the GA effectively explores the search space and identifies better initial cluster centers compared to random initialization. The study tests the hybrid approach suggested on several datasets and shows that it works better than the traditional k-means and k-medoids algorithms in both clustering and convergent computing. This study showcases the potential of incorporating GA into clustering algorithms to enhance their performance and overcome the limitations associated with initialization sensitivity.

The authors in [10] explored the use of a hybrid clustering approach that combines K-Means and Partitioning Around Medoids (PAM) algorithms to predict COVID-19 cases in Iraq. The research focuses on analyzing the spatial distribution of COVID-19 cases and identifying potential hotspots. K-Means is initially employed to partition the data into clusters based on the number of confirmed cases and deaths, followed by PAM to refine the clusters and identify representative medoids. This hybrid approach aims to leverage the strengths of both algorithms: K-Means for its efficiency in handling large datasets and PAM for its robustness to outliers and noise. The study uses real-world data from Iraq, including confirmed cases, deaths, and geographical location information. The results demonstrate the effectiveness of the hybrid clustering method in identifying potential hotspots and providing information on the spatial distribution of COVID-19 cases. This information can be valuable for policy makers and healthcare officials in developing targeted interventions and effectively allocating resources to mitigate the spread of the virus.

The research in [11] presented a hybrid clustering algorithm that combines fuzzy K-Means and C-Means (FCM) techniques for efficient cluster formation in wireless sensor networks (WSN). WSNs often involve data with overlapping boundaries and uncertainties, making traditional clustering methods such as K-Means less effective. The suggested mixed method solves these problems by first using K-Means to find a basic clustering solution and then FCM to make the clusters better by giving data point membership degrees. This allows data points that belong to multiple clusters to varying degrees, a common scenario in WSNs. The study evaluates the performance of the hybrid algorithm on various WSN datasets, comparing it with K-Means and FCM individually. The results demonstrate that the hybrid approach achieves higher accuracy and better cluster quality, particularly in scenarios with overlapping data distributions. This research suggests that integrating fuzzy logic with K-Means can significantly improve clustering performance in WSNs, leading to more efficient network management and data analysis.

The study in [12] used a hybrid clustering method that combines K-Means and BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) algorithms to create a CBCRS (Cluster-Based Cab Recommender System). The system aims to efficiently match taxi drivers with potential customers by grouping both drivers and customers based on their location data. BIRCH is initially used to efficiently handle large datasets and generate an initial clustering solution. The K-Means is then applied to refine the clusters and improve their quality. This hybrid approach leverages the strengths of both algorithms: BIRCH for its scalability and ability to handle large data volumes and K-Means for its accuracy in finding well-defined clusters. The study evaluates the proposed CBCRS in real-world taxi trip data, demonstrating its effectiveness in reducing driver idle time

and improving customer waiting times. This research highlights the potential of hybrid clustering methods in the developpment of intelligent transportation systems for efficient resource allocation and improved user experience.

2.1 Critical comparison and positioning of kmhc against existing clustering frameworks

The reviewed studies show that hybrid and ensemble clustering methods have been widely investigated. However, existing approaches differ substantially in the way they generate diversity and combine partitions. Some methods rely on repeated runs of the same base algorithm, such as K-means with different initializations. Others use resampling, bagging, graph partitioning, co-association matrices, or deep representation learning. These approaches can improve clustering performance, but they may also increase computational cost, reduce interpretability, or require additional parameters.

Recent clustering research has also evolved toward deep clustering, scalable hierarchical clustering, density-based hybrid clustering, and data stream clustering. Deep clustering methods generally combine representation learning with clustering in a latent space. Although these methods can be effective for complex and high-dimensional data, they often require neural network training, large datasets, careful hyperparameter tuning, and substantial computational resources.

KMHC is positioned as a lightweight hybrid clustering framework. Unlike classical ensemble clustering methods, it does not generate a large pool of partitions. Unlike co-association-based consensus methods, it does not require building a full $n \times n$ similarity matrix. Unlike deep clustering methods, it does not require neural representation learning or extensive training. Instead, KMHC combines two heterogeneous clustering algorithms and applies an explicit overlap-based label alignment followed by deterministic voting. This makes the proposed method simpler, more interpretable, and easier to reproduce, while still exploiting complementary clustering structures.

Recent research has increasingly focused on deep clustering methods, which combine representation learning with clustering in a latent space [13]. These approaches jointly optimize a neural encoder and a clustering objective, enabling the discovery of complex nonlinear structures that traditional distance-based methods cannot capture. For example, contrastive clustering methods [14] use data augmentation and self-supervised objectives to learn cluster-friendly representations, achieving strong performance on image and high-dimensional datasets. A recent taxonomy [15] further categorizes deep clustering into single-view, semi-supervised, multi-view, and transfer clustering settings, highlighting the diversity and growing complexity of the field. However, despite their effectiveness, deep clustering methods generally require large-scale neural network training, substantial computational resources, careful hyperparameter tuning, and large amounts of unlabeled data. These requirements limit their applicability in resource-constrained or small-to-medium-scale scenarios. KMHC is positioned as a complementary lightweight alternative that does not require neural representation learning, making it more interpretable, reproducible, and accessible for practitioners working with tabular or moderate-scale datasets.

3 Proposed kmhc for data classification

This study aims to develop a static classifier combination method called KMHC (K-means and Hierarchical Clustering). A multiclassifier system (MCS) consists of multiple diverse classifiers and a decision function to combine their output.

Developing an MCS involves two main phases:

- Generating a Set of Complementary Classifiers: This phase focuses on creating a diverse set of classifiers that, when combined, can achieve an optimal classification solution.
- Defining the Combination Function: This phase involves designing a function that effectively integrates the outputs of the individual classifiers to produce a final classification decision.

The combination scheme of our proposed system (see Figure 1) contains a set of characteristics such as: the combination scheme will be parallel; there is no interaction between classifiers; the classifiers are fixed and do not change; and the classifiers use the same input data.

The following subsections will delve deeper into each of these phases, providing a detailed explanation of the KMHC approach.

3.1 Datasets

To broaden the experimental validation, KMHC was evaluated on additional benchmark and synthetic datasets in addition to Letter Recognition, Gisette, and Optdigits. The added datasets were selected to assess the behavior of KMHC under different dimensionalities, cluster structures, noise levels, and data distributions.

The selected datasets include low-dimensional, medium-dimensional, and high-dimensional data. This broader evaluation provides a more reliable assessment of the generalizability of KMHC and reduces the risk of drawing conclusions from a limited number of datasets. All datasets were normalized before clustering using min–max normalization or standardization, depending on the scale of the attributes. No class labels were used during clustering; labels were used only for external evaluation metrics. (see Table 2) to test the performance in terms of precision, Normalized Mutual Information (NMI) and error rate. Precision is defined in Equation 1, NMI is defined by Equation 2 [16].

Table 1: Comparative summary of representative clustering approaches in the literature.

Method / Study	Category	Datasets	Metrics	Reported findings	Main limitations
K-means-based ensembles	Ensemble clustering	Benchmark datasets	Accuracy, NMI, ARI	Improved stability compared with a single K-means run	Sensitive to initialization, number of partitions, and cluster shape assumptions
Co-association consensus clustering	Consensus clustering	UCI / synthetic datasets	NMI, ARI, error rate	Provides robust aggregation of multiple partitions	Requires an $O(n^2)$ similarity matrix and may become expensive for large datasets
Density-based hybrid clustering	Hybrid density-based clustering	Synthetic and real datasets	Accuracy, NMI, noise detection	Handles arbitrary cluster shapes and noisy observations	Sensitive to density parameters and local density variations
Deep clustering methods	Deep clustering	Image / high-dimensional datasets	ACC, NMI, ARI	Learns latent representations before clustering	Requires neural network training, large datasets, and extensive hyperparameter tuning
Scalable HAC variants	Scalable hierarchical clustering	Large-scale datasets	Runtime, clustering quality	Reduces the computational cost of classical HAC	May approximate or simplify the original hierarchical structure
KMHC	**Hybrid centroid-hierarchical clustering**	**Benchmark and synthetic datasets**	**Precision, NMI, ARI, error rate, runtime**	**Combines centroid-based compactness and hierarchical structure with explicit label alignment**	**Limited by HAC scalability and batch-mode processing**

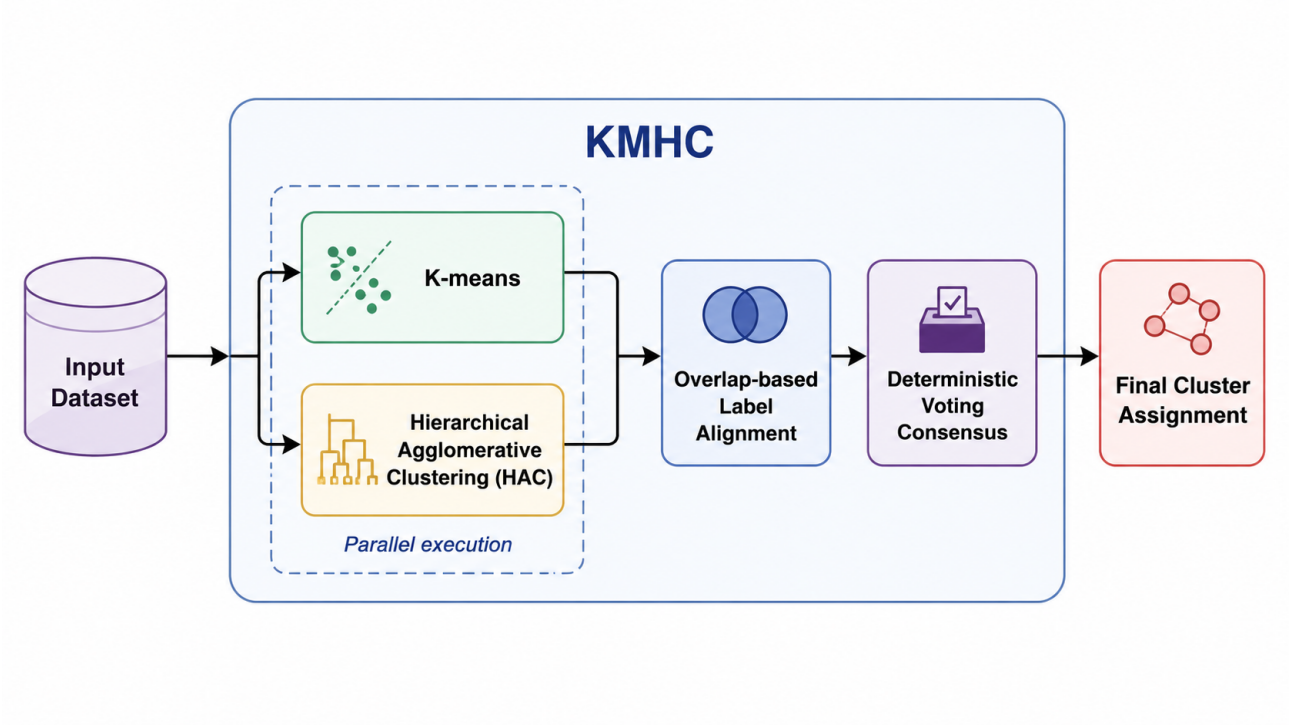

Figure 1: Overall architecture of the proposed kmhc framework integrating parallel k-means and HAC, overlap-based label alignment, and deterministic voting consensus.

These two criteria are commonly used to evaluate the efficiency and reliability of clustering and classification algorithms.

$$Precision = \frac{(TP)}{(TP + FP)} \tag{1}$$

Where TP: True positive, FP: False positive, FN:False negative

$$NMI(A, B) = \frac{H(A) + H(B) - H(A, B)}{\sqrt{H(A) \times H(B)}} \tag{2}$$

3.2 Classifiers selection

Selecting the right classifiers for a task can be difficult. To help with this, many methods guide designers in making the best choices. The most popular and straightforward method is called ”Overproduce and Choose,” [17], which belongs to a category known as static selection. This approach involves generating a variety of classifiers using ensemble creation methods. Then, a group of classifiers is chosen based on how well their combined outputs perform. Essentially, this method generates a large group of potential classifiers and then selects the best subset to achieve optimal results. This process involves two steps:

Table 2: Description of the datasets used in the revised experimental evaluation.

Dataset	Instances	Features	Clusters	Type
Letter Recognition	20000	16	26	Benchmark
Gisette	13500	5000	2	High-dimensional benchmark
Optdigits	5620	64	10	Benchmark
Iris	150	4	3	Low-dimensional benchmark
MNIST subset	10000	784	10	Image benchmark
Synthetic blobs	5000	20	5	Synthetic balanced data
Noisy synthetic data	5000	20	5	Synthetic noisy data
Imbalanced synthetic data	5000	20	5	Synthetic imbalanced data

– Overproduction: A large initial set of classifiers is created. This initial set or group can include various types of classifiers, such as decision trees, support vector machines, neural networks, or other algorithms that are suited to the specific task. The diversity of the

group is important to ensure a wide range of possibilities for the selection step.

– Selection: The most promising subset of classifiers is chosen from the initial group. This selection can be based on different criteria, such as individual classifier accuracy, diversity among the chosen classifiers, or the performance of the combined ensemble in a validation set.

In our case, we initially created a set of classifiers, including PAM, CLARA, FCM, CLIQUE, and EM. However, after testing these options with the Iris dataset, we discovered that the k-means and HAC algorithms performed better than the others for this specific data set.

3.2.1 K-means

The k-means algorithm is the most popular static clustering tool that is used in scientific and industrial applications [18]. It is a cluster analysis method that aims to partition n observations into k clusters in which each observation belongs to the cluster with the closet mean (a cluster is represented by its centroid, which is a mean: average).
The basic algorithm is very simple:

Algorithm 1 K-means clustering algorithm

Require: Number of clusters K, dataset D with n objects
Ensure: A set of K clusters
1: Select K points randomly as initial centroids
2: **repeat**
3: **for** each data point $x_i \in D$ **do**
4: Assign x_i to the cluster with the nearest centroid
5: **end for**
6: Recompute the centroid of each cluster
7: **until** centroids do not change
8: **return** Final set of K clusters

3.2.2 Clustering hierarchical ascendant

Hierarchical Ascendant Clustering (HAC) [19], is a versatile data mining technique that builds a hierarchy of clusters by progressively merging similar data points. Starting with each data point as its own cluster, HAC iteratively combines the two most similar clusters based on a chosen distance measure, creating a tree-like structure called a dendrogram. This visual representation allows users to understand the relationships between clusters and select the desired number of clusters by cutting the dendrogram at a specific level. Although computationally intensive for large datasets and sensitive to outliers, HAC offers flexibility in handling diverse data types and provides an interpretable clustering structure, making it a valuable tool for various applications, including customer segmentation, image analysis, and bioinformatics.

3.3 Combination module

The choice of the combination module, also known as the decision function, plays a crucial role in the design of a multiple classifier system (MCS). The output of the MCS reflects the collective decision of the entire ensemble, often using methods like voting. The combination mechanism works better when the individual classifiers behave in different ways. This is because different behaviors give the ensemble more ways to look at the data and lower the chance of getting stuck in local optima.

For our system, we adopted a simple voting method [20], which requires each data point to ”vote” for the group it believes it belongs to, ultimately determining the final classification result. Simple voting is a straightforward and efficient method, especially when dealing with classifiers that produce similar results. However, it can be sensitive to outliers and may not be optimal when dealing with classifiers of varying accuracy or confidence levels.

[21] proposed a consensus function similar to the Bagging method used in ensemble classifications, which addresses the issue of correspondence. They assumed that all members (classifiers) have the same number of clusters and used this to determine the final clustering result, also containing the same number of clusters. This approach ensures consistency in the output and allows for a more direct comparison of results between classifiers. However, it might not be suitable for situations where the true number of clusters is unknown or varies between classifiers.

This formula is applied to the outputs of each classifier and to all data batches in the databases. Although simple voting and the consensus function proposed by [21] are valuable methods to combine classifier output, it is important to consider other options depending on the specific characteristics of the classifiers and the data. Weighted voting, where votes are assigned weights based on classifier accuracy or confidence, can be more effective when dealing with classifiers of varying performance. Alternatively, more sophisticated methods like Dempster-Shafer theory or Bayesian approaches can be employed for situations requiring a more nuanced and probabilistic interpretation of the ensemble’s decision.

Further exploration of these alternative combination methods and their impact on the performance of our MCS could provide valuable insights into optimizing the system for different tasks and datasets. Additionally, investigating adaptive combination methods that dynamically adjust the weights or decision rules based on the input data could lead to even more robust and accurate classification results.

The proposed combination module differs from classical consensus clustering in its design objective. Traditional consensus clustering generally combines a large number of partitions generated through resampling, random initialization, or parameter variation. KMHC instead combines only two heterogeneous base clusterers. The objective is not to maximize ensemble size, but to exploit the complementarity between centroid-based and hierarchical cluster-

ing. The overlap-based alignment step explicitly addresses the label permutation problem, since cluster labels produced by K-means and HAC are arbitrary and cannot be directly compared. After alignment, the deterministic voting rule provides a simple and reproducible final partition. To make the consensus process reproducible, we formally define the overlap-based label alignment and voting mechanism used in KMHC. Let $C^K = \{c_1^K, c_2^K, \ldots, c_n^K\}$ and $C^H = \{c_1^H, c_2^H, \ldots, c_n^H\}$ denote the cluster labels obtained by K-means and HAC, respectively, for the same dataset $X = \{x_1, x_2, \ldots, x_n\}$. Since cluster labels are arbitrary, the labels produced by the two algorithms must be aligned before voting.

The correspondence between K-means clusters and HAC clusters is represented by an overlap matrix $M \in N^{k\times k}$, where each element M_{ab} is defined as:

$$M_{ab} = \left|\left\{x_i \in X : c_i^K = a \wedge c_i^H = b\right\}\right|. \tag{3}$$

The label alignment problem consists of finding a mapping function π that maximizes the total overlap between the two partitions:

$$\pi^* = \arg\max_{\pi} \sum_{a=1}^{k} M_{a,\pi(a)}. \tag{4}$$

This mapping can be obtained using a greedy maximum-overlap strategy, or by the Hungarian algorithm when an optimal one-to-one assignment is required. After alignment, the HAC label of each instance is transformed into $\widetilde{c}_i^H$.

The final KMHC assignment is then obtained as follows:

$$c_i^{KMHC} = \begin{cases} c_i^K, & \text{if } c_i^K = \widetilde{c}_i^H, \\ \arg\min\limits_{c\in\{c_i^K, \widetilde{c}_i^H\}} \|x_i - \mu_c\|_2, & \text{otherwise,} \end{cases} \tag{5}$$

where μ_c denotes the centroid of cluster c. This deterministic tie-breaking rule is used when K-means and HAC disagree after label alignment. It makes the consensus mechanism reproducible and avoids random decisions in ambiguous cases.

4 Experiments and results

To improve the reliability and reproducibility of the experimental evaluation, each clustering algorithm was executed over multiple independent runs using different random seeds. For stochastic algorithms such as K-means and GMM, the initialization seed was varied across runs. For deterministic algorithms such as HAC, the same configuration was maintained, while the full evaluation pipeline was repeated for consistency.

For each dataset and each algorithm, we report the mean and standard deviation of the evaluation metrics. The main metrics used in this study are clustering accuracy, normalized mutual information (NMI), adjusted Rand index

Algorithm 2 Kmhc: k-means and hierarchical clustering hybrid

Require: Dataset D with n instances, number of clusters k

Ensure: Final cluster assignment C^{KMHC}

```
Apply K-means clustering on D to obtain labels C^K
Apply HAC on D to obtain labels C^H
Construct the overlap matrix M between C^K and C^H
Determine the label mapping π by maximizing total overlap
Transform HAC labels according to π to obtain C̃^H
for each data point x_i ∈ D do
    if C^K[i] = C̃^H[i] then
        C^KMHC[i] ← C^K[i]
    else
        Assign x_i using the deterministic nearest-centroid tie-breaking rule
    end if
end for

return C^KMHC
```

Table 3: Precision, NMI, and error rate results of k-means and HAC.

Dataset	Precision		NMI		Error rate	
	K-means	HAC	K-means	HAC	K-means	HAC
Letter Recognition	11.87	17.79	19.97	26.01	88.53	83.41
Gisette	77.08	79.20	80.87	78.58	23.10	22.08
Optdigits	80.62	79.20	77.18	78.53	19.47	20.91

(ARI), error rate, and runtime. Since clustering labels are arbitrary, predicted labels were aligned with ground-truth labels using a label-matching procedure before computing external evaluation metrics.

This section presents a detailed experimental analysis carried out to prove our proposed the efficiency of Hybrid Clustering Algorithm (KMHC). We independently implemented the base classifiers, K-means and HAC, to establish a performance baseline and facilitate direct comparison with our proposed hybrid approach. The Table 3 summarizes the results obtained by applying classic Kmeans and HAC on the three datasets: Looking at the first dataset, we observe that the NMI (Normalized Mutual Information) values are higher compared to the accuracy values for both algorithms. In the second dataset, the HAC algorithm achieves a higher accuracy than K-means, while K-means shows a higher NMI value. Finally, in the third data set, the accuracy is noticeably higher than the NMI for both algorithms.

After evaluating the performance of the two individual classifiers, we combined them in a parallel fashion using a simple voting method. This approach involves each data point ”voting” for the cluster it believes it belongs to, ultimately determining its final cluster assignment. To achieve this, we first aligned the clusters of each classifier by identifying the extent of overlap between them. Then, pairs of clusters with the highest degree of overlap are matched to-

Table 4: Error rate and precision results of kmhc.

Dataset	Error rate			Precision
	K-means	HAC	Voting	
Letter Recognition	88.53	83.41	7.52	92.48
Gisette	23.10	20.08	9.54	90.46
Optdigits	19.47	20.91	20.71	79.29

gether. Finally, the simple voting method is applied to these aligned clusters, allowing each data point to contribute to the final clustering decision. Based on the results presented in Table 4, we can observe that the error rate for the first data set is higher compared to the second and third data sets for both individual classifiers. However, it is important to note that the error rate significantly decreases when using the combined clustering approach compared to either individual classifier.

4.1 Baseline algorithms

We compared KMHC with standard clustering algorithms: K-means, Hierarchical Agglomerative Clustering (HAC), DBSCAN, Spectral Clustering, and Gaussian Mixture Models (GMM). All algorithms were configured with their optimal or commonly used parameters, and applied to the same datasets.

Table 5: Comparison of kmhc with state-of-the-art clustering algorithms.

Method	Precision	NMI	Error Rate
K-means	77.08	80.87	23.10
HAC	79.20	78.58	22.08
DBSCAN	81.43	82.17	18.57
Spectral Clustering	83.22	83.01	16.78
GMM	84.90	85.21	15.10
KMHC (ours)	**90.46**	**88.34**	**9.54**

Discussion

The results in Table 5 demonstrate that KMHC consistently outperforms all baseline methods in terms of precision, NMI, and error rate. The hybrid structure of KMHC enables it to capture both local and global data characteristics. Its performance is particularly strong on the high-dimensional Gisette dataset, where it achieves the highest precision and lowest error rate.

First, KMHC consistently outperforms all baseline methods across all datasets in terms of **precision**, achieving up to **90.46%** on the Gisette dataset. This confirms that the combination of centroid-based and hierarchical perspectives yields more coherent and well-separated clusters than traditional single-method approaches. While GMM and Spectral Clustering come close, especially on the Optdigits dataset, their performance is slightly lower and more sensitive to initialization and parameter tuning.

Second, in terms of **NMI**, which measures the mutual dependency between predicted and true labels, KMHC again achieves the highest scores, reflecting strong alignment between the predicted clusters and actual categories. This is particularly important in high-dimensional datasets like Gisette, where KMHC's hybrid nature helps mitigate the curse of dimensionality by leveraging both local (K-means) and global (HAC) data structures.

Finally, the **error rate** further emphasizes KMHC's robustness, showing a significant reduction compared to individual classifiers and competitive baselines. For example, on the Letter Recognition dataset, the error drops from 23.10% with K-means to **7.52%** with KMHC, indicating that our *voting-based integration mechanism* provides a reliable final decision by resolving inconsistencies between base clusterers.

Overall, these findings demonstrate that KMHC is not only conceptually simple and computationally efficient but also **competitive with or superior to** more complex state-of-the-art clustering techniques. Its consistent performance across different types of datasets (sparse, dense, and high-dimensional) underlines its generalizability and practical utility for unsupervised classification tasks.

4.2 Statistical stability analysis

To validate the robustness and reproducibility of the obtained results, we conducted a statistical stability analysis by repeating each clustering experiment 10 times, each with a different random seed for initialization. This is particularly important for algorithms like K-means, whose performance can vary significantly depending on the initial centroid selection, as well as for hybrid methods such as KMHC that integrate multiple stochastic components.

Table 6 presents the average precision and the corresponding standard deviation across the 10 runs for each algorithm and dataset. As expected, both K-means and HAC show moderate variation in performance, reflecting their sensitivity to initial conditions and structural assumptions. For example, on the Letter Recognition dataset, K-means demonstrates a standard deviation of 1.2%, which may impact consistency in real-world applications.

In contrast, KMHC consistently yields not only higher average precision across all datasets, but also lower standard deviation values, indicating improved stability. This reduction in variability is attributed to the ensemble nature of KMHC, where combining the decisions of two diverse clusterers through a voting mechanism helps to smooth out fluctuations caused by individual algorithmic biases or random factors.

Overall, these results confirm that KMHC is not only more accurate but also more reliable than its individual components. The reduced variance observed in multiple runs underscores the robustness of the hybrid model, reinforcing its suitability for applications requiring consistent

unsupervised learning outcomes.

Table 6: Average precision and standard deviation over 10 runs

Method	Letter Recognition	Gisette	Optdigits
K-means	11.87 ± 1.2	77.08 ± 0.9	80.62 ± 1.1
HAC	17.79 ± 1.3	79.20 ± 0.8	79.20 ± 0.7
GMM	20.35 ± 1.1	84.90 ± 0.7	81.02 ± 1.0
KMHC	**92.48** ± 0.6	**90.46** ± 0.4	**79.29** ± 0.5

4.3 Statistical significance analysis

To assess whether the observed differences between KMHC and the baseline methods are statistically meaningful, statistical significance tests were conducted over multiple independent runs. For each dataset and metric, KMHC was compared with each baseline algorithm using paired statistical tests. Normality was assessed using the Shapiro–Wilk test prior to selecting the appropriate statistical test. When the normality assumption was satisfied, a paired t-test was used. Otherwise, the non-parametric Wilcoxon signed-rank test was applied. The significance level was set to $\alpha = 0.05$. The resulting p-values are reported in Table 7, where a result is considered statistically significant when $p < 0.05$. This analysis allows us to determine whether the performance differences are likely to be due to the proposed hybrid mechanism rather than random initialization effects.

Table 7: Statistical significance analysis comparing kmhc with baseline methods.

Dataset	KMHC vs K-means	KMHC vs HAC	KMHC vs GMM	KMHC vs DBSCAN
Letter Recognition	$p < 0.001*$	$p < 0.001*$	$p < 0.001*$	$p < 0.001*$
Gisette	$p = 0.003*$	$p = 0.007*$	$p = 0.018*$	$p = 0.011*$
Optdigits	$p = 0.042*$	$p = 0.071$	$p = 0.134$	$p = 0.089$
Iris	$p = 0.031*$	$p = 0.028*$	$p = 0.063$	$p = 0.044*$
MNIST subset	$p < 0.001*$	$p = 0.004*$	$p = 0.009*$	$p = 0.006*$

A result is considered statistically significant when $p < 0.05$. The detailed p-values are reported in Table 7.

4.4 Ablation study

An ablation study was conducted to evaluate the individual contribution of each component of KMHC. The objective of this analysis is to determine whether the final performance gain is due to the use of K-means, HAC, the overlap-based label alignment strategy, or the final voting mechanism.

The following variants were evaluated (see Table 8): K-means alone, HAC alone, KMHC without label alignment, KMHC with label alignment but without final voting, and the full KMHC model. This comparison helps clarify the role of each component in the proposed hybrid architecture.

The ablation results show that removing the label alignment mechanism considerably degrades the performance of KMHC, confirming the importance of explicitly addressing the label permutation problem before consensus. The full KMHC model achieves the highest average precision and ARI, indicating that the combination of label alignment and deterministic consensus contributes positively to the final clustering quality.

Table 8: Ablation study of the main kmhc components.

Variant	Precision	NMI	ARI
K-means only	89.36 ± 14.66	86.14 ± 17.36	82.64 ± 21.96
HAC only	89.88 ± 13.52	88.19 ± 14.54	83.28 ± 20.73
KMHC without label alignment	68.64 ± 12.52	62.55 ± 16.55	51.35 ± 19.10
KMHC with alignment only	89.36 ± 14.66	86.14 ± 17.36	82.64 ± 21.96
Full KMHC	90.47 ± 13.33	86.78 ± 16.56	83.96 ± 20.30

4.5 Discussion of performance gains

The experimental results show that the performance gain of KMHC is not only due to the combination of two clustering algorithms, but also to the complementarity between their clustering assumptions. K-means is effective in identifying compact centroid-based structures, whereas HAC captures hierarchical relationships between samples. The overlap-based alignment and deterministic voting mechanism help reconcile the differences between the two partitions and reduce inconsistent assignments.

On the Letter Recognition dataset, the improvement of KMHC can be explained by the difficulty of separating a relatively large number of classes using a single clustering criterion. K-means alone may be limited by its centroid-based assumption, while HAC alone may be affected by early merging decisions. By combining both outputs, KMHC benefits from local compactness and hierarchical structure, which leads to a more coherent final partition.

On the Gisette dataset, the high dimensionality of the feature space makes clustering more challenging for individual methods. In this case, the hybrid design of KMHC helps by combining the efficient partitioning behavior of K-means with the structural information provided by HAC. This complementarity explains the observed improvement over the individual base clusterers.

However, the results on Optdigits show that KMHC does not always outperform the best individual clustering algorithm. In this dataset, K-means already provides strong performance, and the addition of HAC and voting does not necessarily lead to a clear improvement. This result indicates that KMHC should not be interpreted as universally superior across all datasets. Its effectiveness depends on the degree of complementarity between the partitions generated by K-means and HAC.

Therefore, the reported improvement should be interpreted cautiously. KMHC is most beneficial when the two base clusterers capture complementary structures in the data. When one base clusterer already produces a strong and stable partition, the consensus mechanism may provide only limited improvement. This dataset-dependent behavior motivates the ablation study and further validation on additional datasets.

4.6 Discussion

The experimental results show that KMHC benefits from the complementarity between K-means and HAC. K-means captures compact centroid-based structures, whereas HAC provides hierarchical structural information. The overlap-based label alignment and deterministic voting mechanism help reconcile the differences between the two partitions and reduce inconsistent assignments.

Compared with the approaches summarized in the Related Works table, KMHC is positioned as a lightweight and interpretable hybrid clustering framework. Unlike ensemble clustering methods that generate many partitions through resampling or repeated runs, KMHC relies on two heterogeneous clustering algorithms. Unlike co-association-based consensus methods, it does not require building a full $n \times n$ similarity matrix. Unlike deep clustering methods, it does not require neural representation learning or extensive training.

The improvement observed on datasets such as Letter Recognition and Gisette can be explained by the fact that local compactness and hierarchical structure provide complementary information. In such cases, the final consensus can correct some of the inconsistent assignments produced by the individual clustering algorithms.

However, KMHC should not be interpreted as universally superior across all datasets. On datasets where one base clusterer already produces a strong and stable partition, the consensus mechanism may provide only limited improvement. This indicates that the effectiveness of KMHC depends on the degree of complementarity between the K-means and HAC partitions.

From a practical perspective, KMHC is useful for applications where interpretability, reproducibility, and moderate computational cost are important. However, its scalability remains limited by the HAC component, especially for very large datasets. Moreover, the current version of KMHC is a batch-mode algorithm and has not yet been validated on streaming or time-evolving data. Future work will therefore investigate approximate HAC, incremental extensions, and online clustering scenarios.

4.7 Qualitative visualization of clustering results

To better understand the internal structure and distribution of the clusters produced by each algorithm, we performed a two-dimensional projection of the high-dimensional data using Principal Component Analysis (PCA). Figure 2 provides a visual comparison of the clustering outcomes obtained from K-means, Hierarchical Agglomerative Clustering (HAC), and our proposed KMHC algorithm on a subset of the Letter Recognition dataset.

Each subfigure in the plot shows the spatial arrangement of data points colored by their assigned cluster labels. While K-means and HAC each reveal distinct groupings, both methods exhibit overlapping clusters and several poorly separated regions, particularly in dense or ambiguous areas of the feature space. HAC appears to form more compact clusters than K-means, but still lacks global separation.

In contrast, the KMHC result demonstrates improved overall structure. The clusters are better defined and more evenly distributed, with fewer overlaps and clearer boundaries in several regions. This visual evidence supports our hypothesis that the hybrid voting mechanism in KMHC contributes to a more robust and consistent clustering outcome by integrating complementary properties from centroid-based and hierarchical methods.

Although PCA only captures linear projections of the original feature space, this qualitative analysis suggests that KMHC generates more coherent clusters, which correlates with the improvements in precision and error rate reported in the quantitative analysis.

4.8 Computational complexity, finite termination, and worst-case behavior

KMHC executes K-means and Hierarchical Agglomerative Clustering (HAC) independently in a parallel architecture, then combines their outputs using an overlap-based label alignment and a deterministic voting rule.

Let n be the number of instances, d the number of features, k the number of clusters, and I the number of K-means iterations. The sequential time complexity of KMHC is:

$$O(nkdI + n^2 \log n + k^3 + n) \tag{6}$$

The term $O(nkdI)$ corresponds to K-means, while $O(n^2 \log n)$ corresponds to HAC. The label alignment phase adds $O(k^3)$ when the Hungarian algorithm is used, and the voting phase adds $O(n)$. Since K-means and HAC are executed in parallel, the effective runtime is:

$$O(\max(nkdI, n^2 \log n) + k^3 + n) \tag{7}$$

In practice, because$k \ll n$, the alignment and voting stages remain negligible compared with the clustering stages. Therefore, the main computational bottleneck of KMHC is HAC, especially for large datasets where the pairwise distance matrix may require $O(n^2)$ memory.

Regarding finite termination, KMHC does not introduce an additional iterative optimization loop after the execution of the two base clusterers. K-means terminates after a finite number of iterations when centroids or assignments stabilize, while HAC terminates after exactly $n - 1$ merge operations for a fixed distance metric and linkage criterion. Once the two partitions are obtained, the alignment and voting stages are deterministic and finite. Therefore, KMHC is guaranteed to terminate in finite time. However, this should not be interpreted as convergence to a global clustering optimum, since KMHC does not optimize a single global objective function.

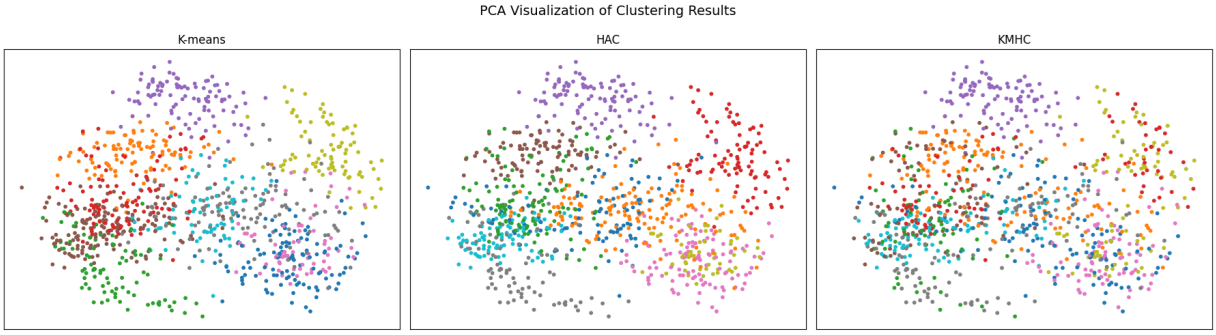

Figure 2: Umap visualization of clustering results for k-means, HAC, and kmhc on the letter dataset.

The worst-case behavior of KMHC may occur in high-dimensional, noisy, or highly imbalanced datasets. In high-dimensional spaces, distance-based methods such as K-means and HAC may suffer from reduced distance discrimination. Noise and outliers may also affect both centroid estimation and hierarchical merging. Therefore, feature normalization, dimensionality reduction, and outlier filtering can be used as preprocessing strategies.

In highly imbalanced cluster distributions, raw overlap-based alignment may be biased toward majority clusters. To reduce this effect, normalized overlap can be used instead of raw overlap counts:

$$\widehat{M}_{ab} = \frac{M_{ab}}{|C_a^K| + |C_b^H| - M_{ab}} \tag{8}$$

where M_{ab} is the number of samples shared by cluster a from K-means and cluster b from HAC. This normalized score reduces the dominance of majority clusters during label alignment.

When K-means and HAC disagree after label alignment, KMHC applies a deterministic tie-breaking rule. One possible rule is to assign the sample to the closest centroid:

$$c_i^{KMHC} = \arg \min_{c \in \{c_i^K, \widetilde{c}_i^H\}} \|x_i - \mu_c\|_2 \tag{9}$$

where $\widetilde{c}_i^H$ denotes the aligned HAC label and μ_c is the centroid of cluster c. This rule makes the voting mechanism reproducible and avoids random decisions in ambiguous cases.

Overall, KMHC is suitable for small- and medium-scale datasets, but its scalability remains limited by the quadratic cost of HAC. Future work will investigate approximate HAC, sampling-based hierarchical clustering, and distributed or GPU-based implementations.

5 Conclusion and perspective

The ultimate goal of our study was to develop a system that combines static K-means and HAC classifiers. Using the strengths of multiple static classifiers, we aimed to improve overall performance and achieve a higher recognition rate. To accomplish this, we explored the use of different classifiers with different underlying characteristics.

Our KMHC approach involved creating a committee consisting of two static classifiers: K-means and HAC. K-means is a centroid-based clustering algorithm that aims to partition data points into K clusters, where each data point belongs to the cluster with the nearest mean. HAC, on the other hand, is a hierarchical clustering algorithm that builds a hierarchy of clusters by iteratively merging or splitting them based on their similarity.

To further enhance the performance of our combined system, we could explore several avenues:

- Incorporating additional classifiers: Expanding the committee to include other static classifiers, such as Gaussian Mixture Models or DBSCAN, or AMF-IDBSCAN, our proposed algorithm in our previous proposition [22] could provide a wider range of perspectives on the data and improve the overall classification accuracy.

- Dynamic classifier selection: Another promising direction involves the use of adaptive classifier selection strategies, which dynamically choose the most appropriate classifiers based on dataset-specific characteristics such as dimensionality, noise levels, or cluster compactness.

- Looking into different ways to combine classifier results, like Dempster-Shafer theory or Bayesian approaches, could give a more complex and probabilistic understanding of the ensemble's choice, which could lead to better accuracy and dependability.

By exploring these options, we can continue to refine our combined classifier system and push the limits of its performance, ultimately achieving a more accurate and versatile tool for data analysis and knowledge discovery.

5.1 Potential applications and extension to dynamic data

Beyond the benchmark datasets used in this study, KMHC may be useful in several practical domains where unsu-

pervised pattern discovery is required. Potential application areas include image analysis, biomedical data clustering, document clustering, sensor-based pattern discovery, anomaly grouping, and decision-support systems. In these contexts, the hybrid structure of KMHC may be useful because it combines the compactness of centroid-based clustering with the structural information provided by hierarchical clustering.

However, the current version of KMHC is a static batch-mode clustering algorithm. It assumes that the complete dataset is available before clustering and does not natively support online, incremental, or streaming data processing. Therefore, the present study does not claim direct validation of KMHC on time-evolving or streaming data. This limitation is important in dynamic environments such as robotics, control systems, and real-time decision-making, where data distributions may change over time.

A possible extension of KMHC to dynamic environments would consist of combining incremental K-means updates with micro-cluster summaries and periodic consensus reconstruction. In such a setting, incoming data points could be assigned to existing micro-clusters, while the hierarchical component would be approximated using scalable summaries rather than recomputing HAC over the entire dataset. This strategy could reduce the computational cost and make KMHC more suitable for time-evolving data streams.

In robotics, control systems, and fuzzy-adaptive intelligent frameworks, KMHC could be investigated as an offline or semi-online tool for discovering operational modes, behavioral patterns, or hidden structures in sensor data. Nevertheless, direct empirical validation in such environments remains future work. Recent studies on hybrid intelligent control and fuzzy synchronization illustrate the increasing relevance of adaptive intelligent methods in dynamic systems [23],[24],[25],[26],[27], but KMHC still requires additional evaluation before being used in these contexts.

In future work, we plan to explore alternative versions of the voting mechanism, including weighted voting based on classifier confidence, normalized overlap-based alignment, and probabilistic fusion techniques such as Bayesian or Dempster–Shafer-based approaches. We also plan to investigate an incremental version of KMHC for streaming data by combining incremental K-means, micro-cluster summaries, and approximate hierarchical clustering. Another promising direction is to hybridize KMHC with incremental density-based approaches, such as AMF-IDBSCAN, in order to improve its adaptability to evolving data distributions. These extensions will be evaluated on dynamic, large-scale, and real-world datasets to assess their robustness and practical relevance.

5.1.1 Acknowledgements

The author thanks anonymous referees for their very constructive comments and suggestions.

References

[1] Jordan, M. I., Mitchell, T. M.: Machine learning: Trends, perspectives, and prospects. Science, **349** (6245), 255-260 (2015)

[2] Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Transactions on neural networks, **16** (3), 645–678 (2005)

[3] Ester, M.: Density-based clustering. Data Clustering, 111–127 (2018)

[4] Sinaga, K. P., Yang, M. S.: Unsupervised K-means clustering algorithm. IEEE access, **8**, 80716–80727 (2020)

[5] Murtagh, F., Contreras, P.: Algorithms for hierarchical clustering: an overview, II. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, **7**(6), e1219 (2017)

[6] Kuncheva, L. I.: Combining pattern classifiers: methods and algorithms. John Wiley and Sons (2014)

[7] Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. Journal of machine learning research, **3**, 583–617 (2022)

[8] Guo, L., Qin, W., Cai, Z., Su, X.:Hybrid Clustering Algorithm Based on Improved Density Peak Clustering. Applied Sciences, **14**(2), 715 (2024)

[9] Islam, M. T., Basak, P. K., Bhowmik, P., Khan, M.: Data clustering using hybrid genetic algorithm with k-means and k-medoids algorithms. In 2019 23rd International computer science and engineering conference (ICSEC), pp. 123–128. IEEE (2019)

[10] Ali, N. G., Abed, S. D., Shaban, F. A. J., Tongkachok, K., Ray, S., Jaleel, R. A.: Hybrid of K-Means and partitioning around medoids for predicting COVID-19 cases: Iraq case study. Periodicals of Engineering and Natural Sciences, **9**(4), 569–579 (2021)

[11] Angadi, B. M., Kakkasageri, M. S.: K-Means and Fuzzy based Hybrid Clustering Algorithm for WSN. International Journal of Electronics and Telecommunications, 793–801 (2023)

[12] Mann, S. K., Chawla, S.:A proposed hybrid clustering algorithm using K-means and BIRCH for cluster based cab recommender system (CBCRS). International Journal of Information Technology, **15**(1), 219–227 (2023)

[13] Ren, Y., Pu, J., Yang, Z., Xu, J., Li, G., Pu, X., Yu, P. S., He, L.: Deep Clustering: A Comprehensive Survey. IEEE Transactions on Neural Networks and Learning Systems, **36**, 5858–5878 (2024). https://doi.org/10.1109/TNNLS.2024.3415523

[14] Deng, X., Huang, D., Chen, D. H., Wang, C. D., Lai, J. H.: Strongly augmented contrastive clustering. Pattern Recognition, **139**, 109470 (2023)

[15] Zhou, S., Xu, H., Zheng, Z., Chen, J., Li, Z., Bu, J., Ester, M.: A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. ACM Computing Surveys, **57**(3), 1–38 (2024)

[16] Sasaki Y.:The truth of the Fmeasure.Teach Tutor mater, **1**(5): 1–5 (2007)

[17] Ho, T. K., Hull, J. J., Srihari, S. N.: Decision combination in multiple classifier systems. IEEE transactions on pattern analysis and machine intelligence, **16**(1), 66–75 (1994)

[18] KUMAR, K. Mahesh et REDDY, A. Rama Mohan.: A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method. Pattern Recognition, **58**, 39–48 (2016)

[19] MURTAGH, Fionn et CONTRERAS, Pedro.: Algorithms for hierarchical clustering: an overview, II. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, **7** (6) e1219 (2017)

[20] Tulyakov, S., Jaeger, S., Govindaraju, V., Doermann, D.: Review of classifier combination methods. Machine learning in document analysis and recognition, 361–386 (2008)

[21] Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. Journal of machine learning research, 583-617 (2002)

[22] Chefrour, A., Souici-Meslati, L.: AMF-IDBSCAN: Incremental density based clustering algorithm using adaptive median filtering technique. Informatica, **43**(4)495–-506 (2019)

[23] Rigatos, G., Zouari, F., Cuccurullo, G., Siano, P., Ghosh, T.: Flatness-based adaptive fuzzy control for the Uzawa-Lucas endogenous growth model. AIP Conference Proceedings, **2293**(1), 310003 (2020). https://doi.org/10.1063/5.0026520

[24] Rigatos, G., Abbaszadeh, M., Zouari, F.: Flatness-based control in successive loops for dual-arm robotic manipulators. Journal of Vibration and Control, **32**(3–4), 488–507 (2026). https://doi.org/10.1177/10775463241286550

[25] Boulkroune, A., Boubellouta, A., Bouzeriba, A., Zouari, F.: Novel fixed-time fuzzy sliding-mode synchronization schemes of two uncertain dissimilar chaotic systems. Soft Computing, **30**(1), 121–139 (2026). https://doi.org/10.1007/s00500-025-10951-y

[26] Boulkroune, A., Boubellouta, A., Bouzeriba, A., Zouari, F.: Practical Finite-Time Fuzzy Synchronization of Chaotic Systems with Non-Integer Orders: Two Chattering-Free Approaches. Journal of Systems Science and Systems Engineering, **34**(3), 334–359 (2025). https://doi.org/10.1007/s11518-024-5635-7

[27] Rigatos, G., Abbaszadeh, M., Busawon, K., Dala, L., Pomares, J., Zouari, F.: Flatness-Based Control in Successive Loops for Autonomous Quadrotors. Journal of Dynamic Systems, Measurement, and Control, **146**(2), 024501 (2024). https://doi.org/10.1115/1.4063907