A Proposal for an Ensemble of Clustering Models for High-Dimensional Data (Draft 03)

Antonio Carlos Rodríguez Bajo
Universidad Internacional Menéndez Pelayo
100011543@alumnos.uimp.es

***Abstract*—High-dimensional datasets present significant challenges for clustering due to issues such as sparsity, noise, the curse of dimensionality, and the presence of irrelevant or redundant features. Traditional clustering algorithms often struggle to yield robust and accurate partitions in such settings. This paper proposes an ensemble clustering framework tailored for high-dimensional data, integrating unsupervised feature selection, dimensionality reduction, and a diverse set of clustering algorithms and consensus functions. The proposed methodology systematically benchmarks combinations of Infinite Feature Selection (Inf-FS), random projections, multiple clustering algorithms, and ensemble selection strategies across several benchmark high-dimensional datasets. Experimental results demonstrate that this integrated approach substantially improves clustering robustness, stability, and accuracy, as evidenced by gains in external validity indices such as Adjusted Rand Index and accuracy. The impact of feature selection and dimensionality reduction is shown to be data-dependent, emphasizing the importance of adaptive ensemble design. The study further provides a reproducible experimental framework and highlights future directions for optimizing ensemble clustering in high-dimensional scenarios.**

***Index Terms*—Ensemble clustering, high-dimensional data, unsupervised feature selection, random projections.**

I. Introduction

CLUSTERING plays an essential role in data analysis as an unsupervised machine learning technique that seeks to group data into meaningful clusters, where elements within the same group share greater similarity compared to those in other groups. This technique has broad applications in diverse fields, including medicine, biology, civil engineering, market research, and social sciences, among others. Despite its utility, it remains a challenging problem due to the absence of ground truth labels, the sensitivity of algorithms to parameter choices, the wide variety of data distributions, and the difficulty of determining the most appropriate number of groups. Moreover, finding optimal clustering solutions is generally an NP-hard problem, which further increases the complexity [1].

A significant theoretical challenge in clustering is presented in Kleinberg's Impossibility Theorem [2]. This theorem asserts that no clustering algorithm can simultaneously satisfy three desirable properties: scale-invariance (the clustering does not change if all distances are multiplied by a positive constant), richness (every possible partition of the data can be obtained by the algorithm for some distance function), and consistency (if distances within clusters are decreased and distances between clusters are increased, the clustering does not change). This result underscores the inherent difficulty of achieving global optimal clustering solutions, particularly when dealing with real-world data that do not adhere to ideal assumptions.

The challenges of clustering are intensified in high-dimensional spaces due to the "curse of dimensionality" [3]. In this scenario, data points tend to become sparse, and the concept of distance, a fundamental aspect of many clustering algorithms, becomes less meaningful [4]. As the number of dimensions increases, the relative distance between the nearest and farthest points diminishes, leading to poor discrimination of distances. This situation not only complicates the identification of meaningful clusters but also increases computational complexity, making traditional clustering algorithms less effective.

Additionally, high-dimensional data often include irrelevant or noisy features that can confound clustering algorithms. For example, algorithms dependent on Euclidean distance may struggle to differentiate between relevant and irrelevant dimensions, further degrading clustering performance [5]. These issues highlight the importance of identifying relevant features and employing advanced preprocessing techniques, such as feature selection and dimensionality reduction, to improve clustering outcomes.

To address these challenges, ensemble clustering has emerged as a promising approach. This method refers to the process of combining multiple clustering results into a single consensus partition, leveraging the strengths of individual algorithms while mitigating their weaknesses. The key idea is to achieve greater robustness, stability, and accuracy by aggregating diverse clustering results [6]. This approach is especially beneficial in high-dimensional settings, where different algorithms may capture complementary aspects of the data [7]. Furthermore, ensemble methods can incorporate feature selection and dimensionality reduction techniques to generate diverse base clusterings and explore various data subspaces, ultimately enhancing the robustness and effectiveness of the final clustering solution [5].

This study introduces an ensemble clustering framework specifically designed for high-dimensional data. The proposed approach integrates unsupervised feature selection, dimensionality reduction techniques and the generation of diverse base clusterings using multiple algorithms and parameter settings. To further enhance robustness, the framework incorporates ensemble selection strategies to retain only the most diverse and high-quality clusterings before applying state-of-the-art consensus functions. By leveraging these components, the framework aims to address the challenges of sparsity, noise, and computational complexity inherent in high-dimensional data, ultimately delivering a robust, scalable, and accurate clustering solution.

This study systematically benchmarks a wide array of ensemble clustering configurations by exploring diverse combinations of unsupervised feature selection, dimensionality reduction, clustering algorithms, ensemble selection strategies, and consensus functions across multiple high-dimensional benchmark datasets. A novel contribution of this work is the construction and empirical evaluation of hybrid ensembles that simultaneously vary clustering algorithms, feature subsets (using Inf-FS), and random projection-based dimensionality reduction, enabling a comprehensive analysis of their combined effects on clustering robustness and accuracy. The findings demonstrate that integrating both feature selection and random projection with an ensemble of clustering models—particularly when combined with consensus functions improves clustering performance on challenging high-dimensional datasets. However, the impact of these techniques is shown to be data-dependent, highlighting the importance of adaptive ensemble design and rigorous benchmarking for optimal performance. This systematic approach, supported by a reproducible experimental framework, represents a significant advancement over previous studies that typically evaluated only isolated components or limited combinations of ensemble methods

The rest of the paper is structured as follows. Section II provides a comprehensive review of the state of the art, discussing topics such as feature selection, dimensionality reduction techniques, generation of base clusterings, ensemble selection, consensus functions, validation metrics and evaluation. Section III introduces the proposed solution, an ensemble of clustering models designed specifically for high-dimensional data. Section IV presents the experimental results and evaluation, including comparisons with existing methods and analysis of the solution's performance. Finally, Section V concludes the paper by summarizing the findings and outlining potential directions for future research.

II. Review of the State of the Art

The application of ensemble methods in clustering high-dimensional datasets has gained significant attention in recent years due to their potential to improve clustering accuracy, robustness, and stability. These methods leverage multiple clustering solutions to derive a consensus clustering that addresses the inherent challenges of high-dimensional data, such as sparsity, noise, and irrelevant and redundant features.

The concept of cluster ensembles as a knowledge reuse framework, which enables the combination of multiple clustering solutions into a single, robust consensus, was introduced in [6]. A key advantage of this approach is its ability to operate without requiring access to the original data features or the algorithms that produced the initial clusterings; instead, it relies solely on the cluster labels provided by each clustering solution. Three consensus functions—Cluster-based Similarity Partitioning Algorithm (CSPA), HyperGraph Partitioning Algorithm (HGPA), and Meta-CLustering Algorithm (MCLA)—were proposed and demonstrated improved clustering quality and robustness across various datasets, including high-dimensional ones. The use of hypergraph representations facilitates effective integration of multiple clusterings, and the ensemble approach can help mitigate some challenges associated with high-dimensional data.

Probabilistic and statistical approaches for clustering ensembles, including a mixture model of multinomial distributions for consensus clustering, were explored in [8], [9]. Ensemble methods can successfully combine ”weak” clusterings—such as those derived from random subspaces, projections, or random hyperplane splits—to produce a superior overall clustering solution. While the mixture model approach requires specifying parameters like the target number of clusters and its performance can be influenced by the diversity and resolution of base clusterings, empirical results show that even simple or weak base clusterings can yield strong consensus results when properly combined.

A bipartite graph partitioning approach for combining cluster ensembles was introduced in [10], demonstrating improved and robust clustering results. To generate diverse base clusterings for high-dimensional data, random projections were used as one ensemble generation method, providing different views of the data and increasing clustering diversity. Although this approach enhances ensemble diversity, both diversity and quality were noted to impact final performance, and highly diverse ensembles (such as those from random projection) may pose challenges for some methods that rely on correspondence between clusters.

The Random Projection Gaussian Mixture Model (RPGMM) [11] is a model-based clustering algorithm specifically designed for high-dimensional data. RPGMM leverages random projections to reduce the dimensionality of the data, thereby simplifying the estimation of Gaussian mixture models (GMMs) which can be computationally challenging in high dimensions. The algorithm applies GMM clustering to several random low-dimensional projections of the data, then aggregates the results using an ensemble approach to enhance clustering stability and accuracy. This method is particularly effective for datasets where conventional GMMs struggle due to the ”curse of dimensionality,” and it demonstrates improved performance, especially when the original variables are highly correlated or when clusters have complex shapes.

Weighted cluster ensemble methods that assign feature-level weights within clusters produced by subspace clustering algorithms were proposed in [7]. This approach improves ensemble performance by leveraging the diversity of clusterings generated with varying parameters and by embedding feature relevance information into the consensus process. The study highlights the challenge of parameter selection in subspace clustering and demonstrates that ensemble methods can provide robust and accurate consensus clusterings without requiring prior knowledge or validation of input parameters.

Recent research has increasingly focused on the scalability of ensemble clustering methods. As discussed in the survey [12], advanced techniques such as graph-based and pairwise similarity-based approaches have been developed to combine clustering results efficiently while preserving solution quality. The survey highlights the strengths of ensemble clustering in integrating heterogeneous data types and providing robust solutions in the face of data perturbations, attributes that have proven particularly valuable in fields like bioinformatics and cybersecurity. However, the authors acknowledge that scalability remains a significant challenge, especially for similarity-based consensus functions, which often have quadratic complexity with respect to the number of data points.

Practical recent applications of ensemble clustering include the study [13], introducing an ensemble clustering method for assessing air quality monitoring networks in Mexico, showcasing the versatility of these methods in environmental science. This approach combined clustering ensembles to evaluate pollution patterns in major metropolitan areas, highlighting their adaptability to real-world datasets. Specifically, the authors applied principal component analysis, hierarchical clustering, and k-means within an ensemble framework to identify similar and redundant monitoring stations in Mexico's three largest cities. This methodology not only improved the robustness of network assessment but also provided actionable insights for optimizing air quality monitoring, underlining the value of ensemble clustering in complex environmental applications.

Ensemble clustering methods have become a promising approach to address the challenges of high dimensionality frequently associated to bioinformatics data. An example is the single-cell RNA sequencing (scRNA-seq) data analysis [14]. In scRNA-seq, identifying cell types and understanding cellular heterogeneity are critical, but the data are often complex, sparse, and noisy, which can undermine the stability and accuracy of individual clustering algorithms. By generating multiple clustering partitions—through varying gene features, cell samples, or clustering algorithms—and integrating them using strategies such as voting or hypergraph-based aggregation, ensemble clustering leverages the complementary strengths of different methods and compensates for their individual weaknesses. This integration yields more robust, accurate, and stable clustering outcomes, thereby enhancing the identification of cell types and the interpretation of cellular heterogeneity in scRNA-seq datasets.

A. Clustering Ensemble Framework

Clustering ensemble methods seek to combine multiple clustering results into a single, superior partition. Several properties have been proposed as desirable criteria for the effectiveness and reliability of clustering ensemble algorithms [9], [15], [16]:

- Robustness: The ensemble should outperform or, at minimum, match the average performance of individual clustering algorithms, especially in the presence of noise and outliers.
- Consistency: The consensus partition should be similar to the individual clusterings being combined, reflecting the structural agreement among base clusterings.
- Novelty: The ensemble method should be able to discover clustering solutions that are unattainable by any single base clustering algorithm alone, potentially revealing new data structures.
- Stability: The resulting clustering should exhibit low sensitivity to variations in the data, parameter settings, and algorithmic randomness, providing reliable results across runs.

The general pipeline for clustering ensemble methods consists of six key steps [12], [17]–[19], as illustrated in Figure 1:

1) **Preprocessing and Feature selection**:
The initial step involves identifying and selecting the most relevant features from the high-dimensional dataset, possibly utilizing unsupervised feature selection techniques to reduce noise and irrelevant information. This may also include standard data preprocessing steps such as normalization and handling missing values.
2) **Dimensionality Reduction**:
To further mitigate the curse of dimensionality, dimensionality reduction techniques such as Principal Component Analysis (PCA) or random projections methods are applied. These techniques transform the data into lower-dimensional subspaces while preserving as much structural information as possible.
3) **Generation of Base Clusterings**:
Multiple base clustering solutions are generated by applying different clustering algorithms, or the same algorithm with varying parameters (e.g., number of clusters, initialization, distance metrics), on the original or reduced data. Diversity in base clusterings can also be achieved by subsampling data points or features, or by projecting the data onto random subspaces.
4) **Ensemble Selection**:
This is often a beneficial step in the clustering ensemble process, particularly when the pool of base clusterings is large or heterogeneous in quality and diversity. Rather than combining all generated base clusterings, ensemble selection aims to identify and retain a subset of base clusterings that are both high-quality and diverse, thus optimizing the performance of the final consensus solution.
5) **Consensus Function**:
The ensemble framework then combines the diverse set of base clusterings into a single consensus partition using a consensus function. Popular approaches include relabeling and voting-based, co-association/pairwise similarity matrix, graph and hypergraph-based, and median partition / optimization-based, each aggregating the input clusterings in different ways to derive a robust final solution.
6) **Validation Metrics and Evaluation**:
The final clustering solution is evaluated using internal

and/or external validation metrics, such as Silhouette Coefficient, Dunn Index, Adjusted Rand Index (ARI), or Normalized Mutual Information (NMI). These metrics assess the quality, stability, and reliability of the consensus clustering, guiding the selection of optimal configurations and ensemble parameters.

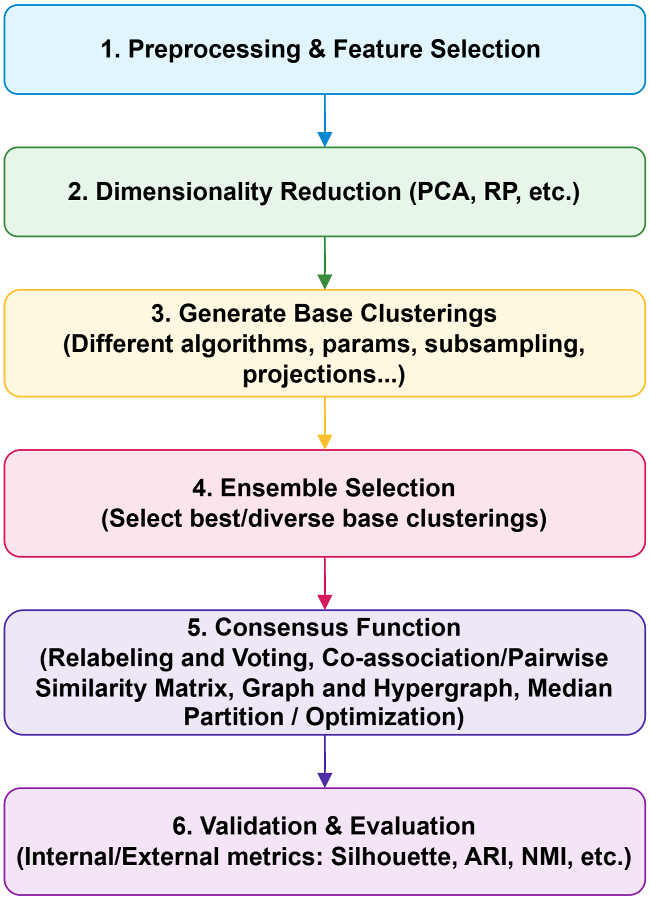
1. Preprocessing & Feature Selection
2. Dimensionality Reduction (PCA, RP, etc.)
3. Generate Base Clusterings
(Different algorithms, params, subsampling, projections...)
4. Ensemble Selection
(Select best/diverse base clusterings)
5. Consensus Function
(Relabeling and Voting, Co-association/Pairwise Similarity Matrix, Graph and Hypergraph, Median Partition / Optimization)
6. Validation & Evaluation
(Internal/External metrics: Silhouette, ARI, NMI, etc.)

Fig. 1. General pipeline for clustering ensemble methods

The following subsections provide a detailed overview of the key aspects involved in the clustering ensemble process.

B. Unsupervised Feature Selection

Unsupervised feature selection aims to identify the most relevant features of high-dimensional data without relying on labeled examples. It is a critical step in unsupervised learning and clustering tasks, as it improves model interpretability and reduces computational costs. Unlike supervised feature selection, where class labels guide the search, unsupervised methods must define “interestingness” or “relevance” directly from the data. Here, “interestingness” refers to how well a subset of features uncovers meaningful cluster structure according to a chosen criterion, while “relevance” denotes the contribution of individual features toward revealing these structures in the absence of class labels [20]. Unsupervised feature selection methods can generally be classified into three main categories: filter, wrapper, and embedded approaches, although some methods incorporate hybrid combinations of these primary types.

Filter methods evaluate features based on intrinsic properties of the data, without relying on the outcome of clustering or classification algorithms. They are fast and scalable, making them suitable for large datasets. These methods often use statistical measures or geometric criteria to assess the importance of features. For example, the Laplacian Score [21] evaluates features by their ability to preserve the local manifold structure of the data. This method constructs a nearest neighbor graph to model the local geometric relationships among data points and computes, for each feature, a score that quantifies how well the feature maintains this locality structure. Features with lower Laplacian Scores are considered more relevant for clustering tasks, as they better preserve locality information and are more likely to capture the underlying intrinsic structure of the data.

Infinite Feature Selection (Inf-FS) [22] graph-based feature selection algorithm that ranks and selects features in unsupervised settings. Inf-FS represents feature subsets as paths in a fully-connected weighted graph, where each node corresponds to a feature and each edge models a pairwise relationship reflecting relevance and redundancy between features. The strength of these relationships is encoded in the adjacency matrix, which can be defined using various criteria—such as variance and correlation for unsupervised Inf-FS.

To evaluate the ”importance” of each feature, Inf-FS considers all possible paths (i.e., feature subsets) of arbitrary lengths within the graph. By leveraging properties of matrix power series and concepts from Markov chains, Inf-FS efficiently computes a feature relevance score by summing the contributions of all paths containing a particular feature, extending the analysis to infinite path lengths. This infinite-path approach allows Inf-FS to efficiently capture high-order dependencies among features (i.e., beyond simple pairwise interactions), while keeping the computational complexity manageable. The final feature ranking is derived from these scores: features that are both individually relevant and collectively non-redundant (i.e., those consistently appearing in high-value paths) are ranked highest.

A significant challenge in feature selection is determining how many features to keep. Inf-FS addresses this with an automatic subset selection mechanism. After scoring all features, Inf-FS analyzes the distribution of feature scores and applies a clustering algorithm (such as 1D Mean-shift) to partition features into groups, typically separating highly relevant features from less informative or redundant ones. The cluster containing the most relevant feature(s) determines the subset to retain

Wrapper methods integrate feature selection directly with clustering algorithms, evaluating feature subsets based on how well they uncover meaningful cluster structure. Although more computationally intensive than filter or embedded methods, wrappers like Feature Subset Selection using Expectation-Maximization clustering (FSSEM) [20] often yield superior results by systematically searching feature subsets and evaluating each via clustering performance criteria such as scatter separability or maximum likelihood. FSSEM uniquely addresses key challenges in unsupervised feature selection: it determines the optimal number of clusters concurrently with feature selection and corrects for biases related to feature subset dimensionality, making it especially robust for high-dimensional data clustering tasks.

Embedded methods integrate feature selection into the learning model itself, allowing simultaneous optimization of feature selection and the learning of clustering or dimensionality reduction structures. This approach leverages the strengths of machine learning models to identify relevant features during training, making them both efficient and effective for unsupervised tasks such as clustering or dimensionality reduction. An example is the Neural Networks and Self-Expression (NNSE) method [23], which combines neural networks and self-expression models in a unified framework. In the context of feature selection, self-expression refers to modeling each feature (or sample) as a linear combination of all the features in the original space, where the learned weights indicate the importance and redundancy of each feature. By minimizing the reconstruction error, the self-expression model highlights the most representative features while filtering out redundant ones. NNSE replaces traditional linear spectral analysis with neural networks to learn nonlinear relationships between data and pseudo labels, uses self-expression to explore feature relationships and select representative features, and employs adaptive graph regularization to preserve the local manifold structure of the original data.

C. Dimensionality Reduction Techniques

Dimensionality reduction is a critical preprocessing strategy for managing high-dimensional datasets, particularly in unsupervised learning scenarios. It involves transforming features from the original dataset to reduce its dimensionality while retaining as much relevant information as possible [24]. This process helps mitigate issues like the curse of dimensionality, noise, and sparsity, which are common in high-dimensional data. The main techniques applied to the clustering ensemble process are explained below.

Singular Value Decomposition (SVD) factorizes a matrix into three components: left singular vectors, singular values, and right singular vectors. This decomposition enables low-rank approximations of the original matrix, which helps in enhancing pattern detection and highlighting dominant trends in the data [25]. SVD has been effectively utilized as a dimensionality reduction technique before clustering, particularly for handling sparse and high-dimensional datasets. However, SVD is computationally expensive, especially for large datasets, and its sensitivity to outliers and nonlinearities can sometimes degrade clustering performance [24].

Principal Component Analysis (PCA) is one of the most widely used linear dimensionality reduction techniques. PCA applies SVD in a specific way to maximize variance and identify principal components, transforming the original data into a set of orthogonal components called principal components, which capture the maximum variance in the dataset. PCA finds the optimal linear projections using eigenvector decomposition, ensuring minimal information loss [26]. Although PCA is utilized in clustering ensembles to reduce data dimensionality, its effectiveness depends on the dataset and clustering method. PCA combined with random subsampling (PCASS) has been studied for ensemble clustering, where it can improve clustering performance by generating diverse ensemble members, though its success may vary across datasets [18].

Random Projection (RP) is a dimensionality reduction technique that maps high-dimensional data onto a lower-dimensional subspace using a randomly generated projection matrix, while approximately preserving distances between data points. Theoretical guarantees, such as the Johnson-Lindenstrauss lemma [27], ensure that pairwise distances are maintained with minimal distortion. In ensemble clustering, random projections are valuable for generating diverse base clusterings by projecting the data onto different random subspaces, which enhances the robustness and quality of the final consensus clustering. These diverse views help capture various aspects of the data structure, mitigating issues like sparsity and high-dimensional noise [11].

D. Generation of Base Clusterings

The generation of base clusterings involves producing multiple clustering results that can later be combined into a consensus solution. The quality and diversity of the base clusterings are critical for the performance of the ensemble clustering method.

To generate diverse base clusterings, several strategies are commonly employed. One approach involves applying different clustering algorithms, such as k-means, hierarchical clustering, spectral clustering, or density-based clustering, to the same dataset to produce varied results. This method leverages the distinct strengths and criteria of each algorithm, enhancing ensemble diversity [28]. Another strategy focuses on parameter variations, where the same algorithm is run with different configurations, such as varying the number of clusters, initialization methods, or distance metrics. This ensures that even a single algorithm can yield diverse clusterings [12].

Subsampling and random projection offer another effective approach by utilizing random subsets of data points or projecting the data onto random subspaces to create diverse clustering outputs [6], [11], [29]. Additionally, ”weak” clustering algorithms, which are simple methods, such as random 1-dimensional projections or random splitting by hyperplanes, that might not perform well individually, are often integrated into ensembles. Their contribution lies in providing diversity, enabling meaningful ensemble results [8], [9].

Ensemble clustering combines a variety of clustering algorithms to leverage their unique strengths and mitigate their individual limitations. Among these, partition-based methods like k-means and k-medoids play a prominent role due to their simplicity and efficiency. These algorithms produce hard partitions of the data, meaning each data point belongs to exactly one cluster. Their computational speed and ease of implementation make them widely used, especially in scenarios where the data conforms to spherical clusters or low-dimensional spaces. However, they may face challenges when applied to high-dimensional or complex data structures [30].

Hierarchical clustering, which constructs dendrograms via single-linkage, complete-linkage, or average-linkage algorithms, is commonly used as one of the base clustering methods in ensemble clustering frameworks. By generating base clusterings with hierarchical algorithms, potentially using different linkage criteria or varying subsets of features or data points, ensemble methods can capture a range of structural patterns and clustering tendencies present in the data. The diversity among base clusterings produced by hierarchical approaches contributes to the robustness and stability of the overall ensemble solution. However, hierarchical clustering can be computationally intensive for large-scale problems, especially when constructing dendrograms for large dataset [12].

Spectral clustering, such as the Ng–Jordan–Weiss method [31], offers a graph-based approach to partitioning data. By leveraging eigenvalue decomposition of similarity matrices, spectral clustering transforms the clustering problem into a graph partitioning challenge. This method is well-suited for non-linearly separable data and excels in detecting clusters with intricate structures. Despite its powerful capabilities, spectral clustering requires careful handling of input parameters and computational resources, particularly for large-scale datasets [28].

Density-based methods such as Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise (DBSCAN) [32] and Ordering Points To Identify the Clustering Structure (OPTICS) [33] focus on identifying clusters based on varying densities. These algorithms are particularly adept at handling data with noise and detecting clusters of arbitrary shapes. Their reliance on density metrics allows them to effectively exclude outliers and uncover patterns in datasets where traditional methods struggle. However, selecting appropriate parameters for density thresholds can be challenging and requires domain expertise [12].

Fuzzy clustering methods, such as Fuzzy C-Means [34], provide a unique approach by allowing data points to belong to multiple clusters with varying degrees of membership. This is particularly useful for ambiguous or uncertain datasets where hard partitions may oversimplify the underlying structure. Fuzzy clustering is effective in capturing nuanced relationships between data points but can be computationally demanding, especially when dealing with high-dimensional data [12], [35].

Model-based clustering methods, such as Gaussian Mixture Models (GMMs), represent another important class of algorithms for generating base clusterings [6]. GMMs assume that data are generated from a mixture of several Gaussian distributions, each corresponding to a cluster, and use statistical methods (typically the Expectation-Maximization algorithm) to estimate the parameters of these distributions. This probabilistic approach allows for soft assignments of data points to clusters and can capture more complex cluster shapes than methods like k-means. GMMs are especially valuable in situations where clusters have different covariance structures or overlap in feature space, though they can be computationally intensive for high-dimensional data and require model selection strategies to determine the optimal number of components.

These diverse clustering algorithms, each with its strengths and weaknesses, can be integrated into the ensemble clustering process to enhance robustness, accuracy, and stability across varying data types and structures.

E. Ensemble Selection

Ensemble selection in clustering ensembles focuses on choosing the best subset of base clusterings to improve consensus results, since including all partitions, especially low-quality or redundant ones, can reduce performance. Selecting clusterings that are both diverse and of high quality can significantly improve the resulting consensus partitions [19].

High quality refers to base clusterings that are internally coherent and capture meaningful structure within the data; in practice, this is often assessed using internal validity measures or by evaluating how well a clustering agrees with the overall trend of the ensemble, such as through the Sum of Normalized Mutual Information (SNMI). Diversity, on the other hand, measures how different the selected clusterings are from each other, ensuring that the ensemble combines complementary perspectives rather than redundant solutions. This is commonly quantified by calculating low pairwise similarity (e.g., low NMI) between clusterings [36].

Recent research has advanced clustering ensemble selection by introducing sophisticated strategies such as hierarchical and multiplex network-based approaches, which utilize ensemble diversity and quality through various validity indices and community detection techniques [37], [38]. Other works have developed selection strategies that explicitly combine internal and external validity measures to guide the process and improve consensus performance [39]. There is also a growing emphasis on adaptive, data-driven criteria for ensemble selection, where the contribution of each candidate partition is evaluated relative to the ensemble in a dynamic and context-aware manner [36]. Furthermore, recent studies highlight the benefits of ensemble selection for computational efficiency, particularly in large-scale and high-dimensional settings, where reducing ensemble size while maintaining accuracy is essential [12].

F. Consensus Functions

A core component of clustering ensemble methods is the consensus function—the strategy that combines multiple base clusterings into a single, robust consensus partition. The choice of consensus function significantly influences the quality, stability, and interpretability of the final clustering solution. Various consensus functions have been developed, which can be broadly categorized based on the type of information they utilize and their computational mechanism.

In the context of clustering ensembles, n denotes the number of data points or objects being clustered, r (also sometimes denoted as m or H) is the number of base clusterings (partitions) in the ensemble, and k is the number of clusters in the final consensus partition. Below, a taxonomy of consensus functions is presented [6], [9], [12], [16].

1) Relabeling and Voting-Based Approaches: These methods address the label correspondence problem by aligning cluster labels across partitions—typically using the Hungarian algorithm [40], which has a computational cost of $O(k^3)$—before assigning each object to the cluster label that receives the most votes (via plurality or majority voting). Such approaches are most effective for small datasets and for ensembles where the number of clusters is fixed across all partitions, allowing reliable and efficient label alignment. However, as the number or diversity of clusters increases, or when base clusterings contain different numbers of clusters, these methods become less practical due to the ill-posed nature and computational complexity of label alignment.

Majority voting is a simple and intuitive example: after label alignment, each data point is assigned to the cluster label it receives most frequently across all base clusterings. This method is most feasible when the number of clusters is fixed and label alignment is reliable; otherwise, ambiguities and computational costs escalate.

The k-modes algorithm is another approach suited to categorical data and categorical cluster assignments. It generalizes k-means clustering to operate directly on categorical variables, grouping cluster assignments based on the mode (most frequent value) rather than the mean. This makes k-modes especially practical for consensus over cluster labels, which are inherently categorical.

2) Co-association / Pairwise Similarity Matrix Methods: These methods construct an $n \times n$ co-association matrix, where each entry (i, j) records the frequency (or proportion) with which objects i and j appear in the same cluster across ensemble members. This matrix serves as a new similarity measure, to which a standard clustering algorithm (such as hierarchical agglomerative clustering) is applied to extract the consensus clustering. Evidence Accumulation Clustering is a prominent example, using the co-association matrix as input for linkage-based clustering to produce the consensus partition. While intuitive and effective for small to medium datasets, this approach has $O(n^2)$ computational and memory complexity, making it unsuitable for very large data sets.

3) Graph and Hypergraph-Based Methods: These approaches represent relationships among data points and clusters as graphs or hypergraphs: nodes correspond to data points (or clusters), and edges or hyperedges encode co-membership or similarity. Consensus clustering is achieved by partitioning this (hyper)graph using graph partitioning algorithms.

The Cluster-based Similarity Partitioning Algorithm (CSPA) constructs a similarity (co-association) matrix where each entry reflects how often two data points are assigned to the same cluster across all partitions. This matrix is interpreted as a weighted adjacency graph, and a graph partitioning algorithm (such as spectral clustering) is applied to derive the consensus partition. CSPA is effective but has a quadratic time and space complexity in the number of data points.

The Linkage Clustering Ensemble (LCE) method constructs a link-based similarity between clusters across partitions, capturing not just pairwise object similarity but also higher-order relationships. It then applies clustering to these links to produce the consensus. LCE is particularly useful when cluster relationships are more informative than direct object co-occurrence, and has been shown to perform well in various applications.

4) Model-Based and Optimization Methods: These methods formulate consensus clustering as an optimization or statistical inference problem, aiming to find the consensus partition that best explains or summarizes the ensemble.

Latent Class Analysis (LCA) is a probabilistic approach that models the observed cluster assignments as being generated by latent (hidden) class variables. LCA infers the most likely consensus cluster labels by estimating the underlying probability distributions, making it well-suited for situations where partitions are noisy or partially observed.

Other optimization-based approaches, such as those using Non-Negative Matrix Factorization (NMF), encode the ensemble into a matrix and factorize it to yield soft assignments, directly targeting the consensus objective. These are theoretically well-founded and efficient for large datasets but can be computationally demanding for very large n or complex similarity metrics.

G. Validation Metrics and Evaluation

Evaluating the quality and effectiveness of clustering ensembles is a fundamental yet challenging task, especially in the absence of ground truth labels, a common scenario in unsupervised learning. The assessment of clustering solutions typically relies on cluster validity indices, which can be broadly categorized into internal, external, and relative indices. Each of these metrics evaluates different aspects of clustering performance and offers complementary insights into the quality of the partitions produced by ensemble methods [12].

Internal validity indices are essential tools for evaluating the quality of clustering solutions without relying on external information such as class labels. These indices assess clustering results by examining the inherent structure of the data, focusing primarily on two key aspects: compactness (how closely related the data points within the same cluster are) and separation (how distinct different clusters are from each other).

Some of the most widely used internal validity indices are the Davies-Bouldin Index, which evaluates the average similarity between each cluster and its most similar one, rewarding compact and well-separated clusters with lower values; the Silhouette Coefficient, which combines measures of both intra-cluster cohesion and inter-cluster separation, with higher average values indicating better clustering; and the Calinski-Harabasz Index and Dunn Index, which rely on ratios of between-cluster and within-cluster distances to quantify cluster definition, each with its own sensitivity to cluster shapes and densities. Other metrics, like the C Index, Gamma Index, PBM Index, Ray-Turi Index, and McClain-Rao Index, introduce alternative formulations for assessing the internal structure of clusters, often considering variances, pairwise distances, or cluster compactness [36].

External validity indices compare clustering results to an external reference or ground truth (when available), providing a direct measure of agreement between the obtained clusters and known class labels. A wide range of external indices have been developed, each capturing different aspects of similarity or agreement. Commonly used indices include the Adjusted Rand Index (ARI) and the Rand Index (RI), which are pair-counting measures that quantify the proportion of sample pairs on which two partitions agree; the Normalized Mutual Information (NMI), Mutual Information (MI), and Adjusted Mutual Information (AMI), which are information-theoretic measures evaluating the shared information between cluster assignments and ground truth; and the Variation of Information (VI) and Entropy, which quantify the information lost or gained between the two partitions.

Set-matching measures such as Purity and F-Measure (F1-score) assess the extent to which clusters contain a single class or the harmonic mean of precision and recall, respectively. Additional pairwise indices like the Jaccard Index and the Fowlkes–Mallows Index also provide valuable perspectives on clustering agreement. More recently, the Chi Index, based on the chi-squared statistical test, has been proposed to directly assess the dependence between clustering and ground truth assignments using contingency tables. [36], [43]

Relative validity indices are designed to compare the quality of different clustering results or parameter settings on the same dataset by quantifying how changes in the clustering structure affect certain quality measures. Unlike absolute indices, which provide an overall assessment of a single clustering, relative indices are particularly useful for evaluating the impact of algorithmic choices, such as the number of clusters, distance metrics, or initialization methods. [36]

Empirical validation in clustering ensemble research frequently relies on combining multiple validity indices to achieve a comprehensive evaluation, particularly for high-dimensional datasets where individual metrics may be inadequate. Benchmarking on both synthetic and real-world datasets, in conjunction with the use of diverse validity indices, is recommended to assess the reliability and generalization of ensemble clustering methods.

III. Solution Proposal

This solution employs ensemble clustering tailored for high-dimensional datasets to enhance robustness, stability, and accuracy. By integrating multiple base clustering results into a unified consensus partition, this approach leverages the strengths of diverse clustering methods, effectively addressing challenges such as sparsity, noise, and the lack of a universally optimal individual clustering method. This methodology was specifically chosen for its suitability to high-dimensional settings and its ability to deliver strong results even under limited computational resources, without relying on complex or highly resource-intensive algorithms.

Fundamentally, the method relies on two key principles: diversity in base clusterings, achieved by varying algorithms, parameters, and feature subsets, and a consensus function, responsible for synthesizing these heterogeneous partitions into a single, cohesive clustering solution.

The six key steps of the proposed solution, as illustrated in Figure 2, are as follows:

1) **Preprocessing and Unsupervised Feature Selection:**
First, standard data preprocessing such as normalization and handling missing values is performed. Then, Infinite Feature Selection (Inf-FS) algorithm is employed as an unsupervised feature selection method. This technique is selected because it automatically determines the optimal number of features to retain, thereby facilitating reproducibility and reducing manual intervention. Feature selection is applied as a first step to mitigate the curse of dimensionality and improve cluster structure.
2) **Random Projections:**
Gaussian random projection is applied to the selected set of features. The projection maps the data onto a lower-dimensional space, with the number of projected dimensions set to one-tenth of the number of retained features by Inf-FS, rounded up to the nearest multiple of 10. To prevent overly large projected spaces and maintain computational efficiency, the number of dimensions is further capped at a maximum of 100. This constraint ensures scalability of the ensemble clustering process, while promoting diversity among base clusterings and enhancing robustness to noise.
The projected dimension d of each random projection is determined based on theoretical and empirical findings regarding the preservation of cluster structure. In particular, projecting high-dimensional features onto a subspace of dimension $d = O(logG)$, where G is the number of clusters, is sufficient to almost perfectly preserve inter-cluster separation for GMMs. The empirical analysis further recommends $d = 10log(G) + 1$ as a practical guideline, noting that increasing d beyond this value does not significantly improve clustering performance [11].
3) **Diverse Clustering Algorithms:**
A comprehensive set of clustering algorithms is employed to generate base clusterings. These included Nonnegative Matrix Factorization (NMF), Hierarchical Clustering (HC), Divisive Analysis Clustering (diana), k-means (KM), Partitioning Around Medoids (PAM), Affinity Propagation (AP), Spectral Clustering (SC), Gaussian Mixture Models (GMM), Self-Organizing Maps (SOM), Fuzzy c-means (CMEANS), and Hierarchical DBSCAN (HDBSCAN).
4) **Consensus Function:**
For consensus formation, the following methods are

applied: kmodes, Majority voting, Linkage-based Cluster Ensembles (LCE), Link-based Cluster Aggregation (LCA), and Cluster-based Similarity Partitioning Algorithm (CSPA).

5) **Validation Metrics and Evaluation:**
For each dataset, the best performing combination of feature selection, projection, clustering algorithm, and consensus function is identified. The evaluation is based on external metrics most widely accepted in the literature for that particular dataset (e.g., Adjusted Rand Index, Normalized Mutual Information, accuracy, etc.), ensuring fair and meaningful comparisons.

6) **Statistical Significance Analysis:**
The statistical significance of the performance improvement is assessed. Specifically, the best ensemble configuration is compared against the other configurations. Since the executions are not strictly paired—due to the changes introduced by the feature selection and/or random projection steps in the pipeline—the appropriate statistical test is the Mann-Whitney U test (also known as the Wilcoxon rank-sum test) for independent samples. This is a non-parametric test, meaning it does not assume that the data follow a normal distribution.
In this study, 20 runs were performed for each configuration. This sample size is widely regarded as a practical minimum for applying the Mann-Whitney U test in benchmarking and machine learning evaluation contexts, offering a balance between computational efficiency and statistical robustness.

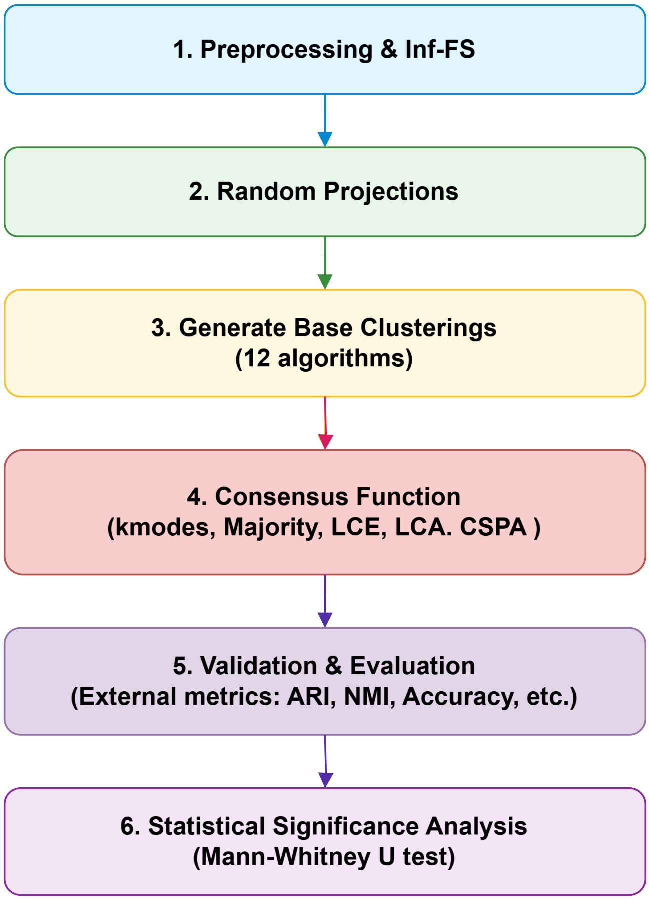
1. Preprocessing & Inf-FS
2. Random Projections
3. Generate Base Clusterings
(12 algorithms)
4. Consensus Function
(kmodes, Majority, LCE, LCA. CSPA)
5. Validation & Evaluation
(External metrics: ARI, NMI, Accuracy, etc.)
6. Statistical Significance Analysis
(Mann-Whitney U test)

Fig. 2. Pipeline for the proposed clustering ensemble method

A. Experimental Framework

The experimental framework was designed to ensure reproducibility, scalability, and extensibility across different computational environments and programming languages.

1) Computing Setup: The experiments were conducted on a computer running Microsoft Windows 11 Home, equipped with an Intel Core i9-14900HX processor (32 cores) and 32 GB of RAM.

2) Programming Tools and Libraries: The diceR package [44], [45] in R served as the primary tool for implementing the ensemble clustering framework. This package provides a comprehensive suite of functions for performing cluster analysis using an ensemble clustering approach. It supports the generation of diverse base clusterings through multiple algorithms, the application of various consensus functions, and the evaluation of clustering performance via both internal and external validity indices. To extend its capabilities, an enhanced version named diceRplus [49] was developed, introducing unsupervised feature selection, dimensionality reduction, and additional functionalities.

Auto-UFSTool [47] was utilized for unsupervised feature selection. This MATLAB toolbox provides a collection of 25 robust unsupervised feature selection approaches, most of which were developed within the last years. It enables the evaluation of feature selection results and generates comparative graphs for different feature subsets. By incorporating Auto-UFSTool into the workflow, the ensemble clustering framework can benefit from state-of-the-art unsupervised feature selection techniques, potentially improving the quality and relevance of the input features for subsequent clustering steps.

R.matlab [48] was used to connect R and MATLAB. It is an R package that allows communication between R and MATLAB via a TCP/IP connection. The package provides methods for reading and writing MAT files, as well as sending commands and data between R and MATLAB.

3) Experimental Design and Reproducibility: To guarantee reproducibility and systematic analysis, an experiment management framework was established. This included:

- Configuration Management: All experiment configurations (algorithm choices, parameter settings, random seeds, data splits) were stored in version-controlled files.
- Result Storage: Outputs—including cluster assignments, consensus matrices, and validity scores—were stored in structured format (dataframe) for traceability and post-hoc analysis.
- Automation & Analytics: Scripts were be developed to automate the running of experiments, aggregation of results, and generation of analytical visualizations.
- Documentation and Code Sharing: All scripts and doc-

umentation were maintained in a git repository [49] to facilitate transparency.

IV. Experimental Results and Evaluation

In this section, the evaluation of the proposed ensemble clustering method is conducted using several benchmark high-dimensional datasets. These datasets were chosen from various domains, such as spectroscopy, genomics, and image analysis, to evaluate the generalizability and robustness of the ensemble clustering framework. Table I provides a summary of the main characteristics of the datasets employed in the experiments, including their domain, a brief description, and their dimensionality.

TABLE I
Summary of Benchmark Datasets Used in Experiments.

Dataset	Domain	Description	k	Samples	Features
Meat [50]	Food	Spectroscopy of raw meat from 5 animal species.	5	231	1,050
Lung Cancer [51]	Genomics	Expression profiles of 4 lung cancer types and normal tissue.	5	203	3,313
Lymphoma [52]	Genomics	Lymphoma gene expression.	3	62	4,026
Prostate GE [53]	Genomics	Microarray data from prostate tumor and normal samples.	2	102	5,966
warpPIE10 [54]	Image	PIE face subset	10	210	4,096

A. Initial Experiments: Validating the Integration of Unsupervised Feature Selection with Ensemble Clustering

To validate the proposed methodology, a series of initial experiments was designed to assess the impact of unsupervised feature selection when integrated with ensemble clustering, with particular focus on the random projection-based RPEClu algorithm [11]. The motivation for these experiments was to determine whether removing irrelevant or redundant features prior to clustering could alleviate key limitations such as sensitivity to noise and substantial computational demands, both of which are common in high-dimensional data analysis. In this setting, the RPEClu approach utilizes random projections to transform the original high-dimensional data into a set of lower-dimensional subspaces. Within each projected subspace, GMMs clustering were applied, and the resulting cluster assignments are combined using consensus clustering. To identify and retain the most informative and non-redundant features before ensemble clustering, Inf-FS feature selection was employed. Furthermore, to improve computational efficiency, a parallelized version of RPEClu was developed for this work, enabling concurrent processing of random projections, thereby significantly reducing execution time while maintaining clustering quality.

1) RPEClu: Original vs. Parallelized Implementation: The original single-threaded RPEClu algorithm ($G = 5$, $d = 17$, $B = 1000$, $B^* = 100$)—where G is the number of clusters, d is the dimension of the random projections, B is the total number of random projections, and B^* is the number of top-performing projections used for consensus—was applied to the meat dataset to reproduce the results reported in [11]. To enhance computational efficiency, the random projection ensemble process was parallelized. This modification significantly reduced processing time while maintaining clustering quality, as shown in Table II.

TABLE II
RPEClu Implementation Performance

RPEClu Implementation	Exec. Time (s)	ARI
Single-threaded as reported in [11]	6,460	0.32
Single-threaded version	2,475	0.29
Parallelized version	356	0.31

2) Impact of Unsupervised Feature Selection: Experiments were conducted using Inf-FS between 25 and $1,050$ variables, increasing in steps of 25. RPEClu algorithm ran with $(G = 5, d = 17)$ and 2 configurations $(B = 1000, B^* = 100$ and $B = 500, B^* = 50)$. The resulting internal clustering metrics, as a function of the number of features selected by Inf-FS, are summarized in Figure 3 .

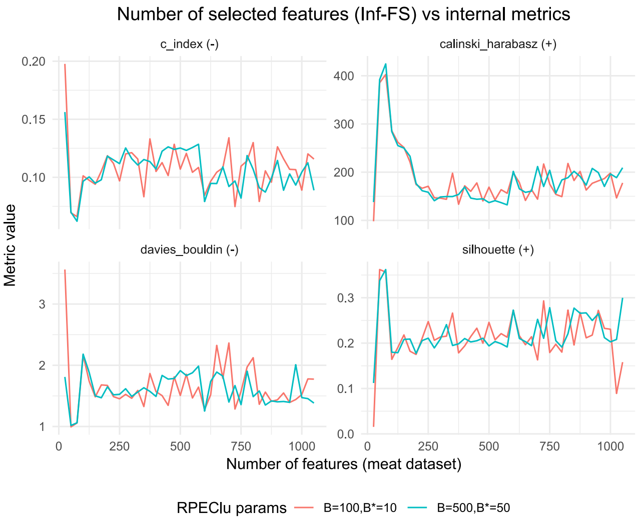

Fig. 3. Internal clustering metrics (y-axis) as a function of the number of features selected by Infinite Feature Selection (x-axis) on the meat dataset, evaluated using RPEClu with two parameter configurations. Higher (+) or lower (-) values indicate better clustering.

These plots illustrate how the performance of RPEClu varies across a range of feature subset sizes, from 25 up to 1,050 variables. Notably, several internal metrics—including silhouette, Calinski-Harabasz, and Davies-Bouldin—reach optimal or near-optimal values at intermediate feature counts (typically between 50 and 100 features), suggesting that excessive dimensionality reduction may eliminate informative variables, while retaining too many features reintroduces noise and redundancy. Overall, these results indicate that careful tuning of the feature selection step is crucial for maximizing clustering performance: an appropriate balance between dimensionality reduction and information preservation leads to more compact, well-separated, and robust clusters. This trend is consistent across both parameter configurations tested $(B = 1000, B^* = 100 \ and \ B = 500, B^* = 50)$, underscoring the stability of the ensemble clustering approach when combined with unsupervised feature selection.

Table III summarizes the best values achieved for each internal clustering metric across the different experimental configurations. Each entry indicates the experimental parameters, the corresponding number of selected features, and the best metric value obtained. Table IV presents a ranking of the experimental configurations that achieved the best values across the various internal validation indices, indicating for each configuration the number of metrics in which it attained the top result as well as the corresponding ARI value. This table allows for the identification of the most robust configurations in terms of consistent performance according to multiple clustering quality metrics. Experiment with parameters $(B = 100, B^* = 50, num\_features = 75)$ stands out as the preferred configuration due to its superior consistency and robustness, achieving top results across four internal metrics. This demonstrates its ability to maintain high clustering quality across diverse evaluation criteria, ensuring stable and reliable outcomes in different scenarios. Additionally, its external metric remains consistently strong $(ARI = 0.68)$

TABLE III
BEST INTERNAL METRICS (INF-FS FEATURE SELECTION - MEAT DATASET)

Metric	Params (B, B^*)	Num Features	Best Value
c_index	500, 50	75	0.06
calinski_harabasz	500, 50	75	424.63
Compactness	1000, 100	25	0.13
Connectivity	500, 50	975	64.48
davies_bouldin	1000, 100	50	1.01
dunn	500, 50	600	0.06
gamma	500, 50	400	10.33
mcclain_rao	500, 50	75	0.27
pbm	500, 50	1050	13.53
ray_turi	1000, 100	50	0.52
sd_dis	1000, 100	25	38.47
silhouette	500, 50	75	0.36
tau	500, 50	575	0.00

Figure 4 illustrates the relationship between the number of selected features by the Inf-FS unsupervised method and the ARI in the meat dataset. The baseline $ARI = 0.32$ reference line marks the ARI achieved without feature selection in [11], providing a benchmark to assess improvements across varying feature subsets. The plot shows substantial gains in clustering performance when feature selection is applied. Both configurations achieve comparable clustering results, but the $B = 500, B^* = 50$ setting offers a reduction in execution time, making it the more efficient choice.

TABLE IV
RANKING OF BEST INTERNAL METRICS (INF-FS FEATURE SELECTION - MEAT DATASET)

Rank	Params (B, B^*)	Feat.	Count	ARI
1	500, 50	75	4	0.68
2	1000, 100	25	2	0.53
3	1000, 100	50	2	0.69
4	1000, 100	75	1	0.69
5	500, 50	400	1	0.44
6	500, 50	575	1	0.61
7	500, 50	600	1	0.37
8	500, 50	675	1	0.54
9	500, 50	975	1	0.30
10	500, 50	1050	1	0.31

Figure 4 illustrates the relationship between the number of selected features by the Inf-FS unsupervised method and the ARI in the meat dataset. The baseline ARI=0.32 reference line marks the ARI achieved without feature selection in [11], providing a benchmark to assess improvements across varying feature subsets. The plot shows substantial gains in clustering performance when feature selection is applied. Both configurations achieve comparable clustering results, but the $B = 500, B^* = 50$ setting offers a reduction in execution time, making it the more efficient choice.

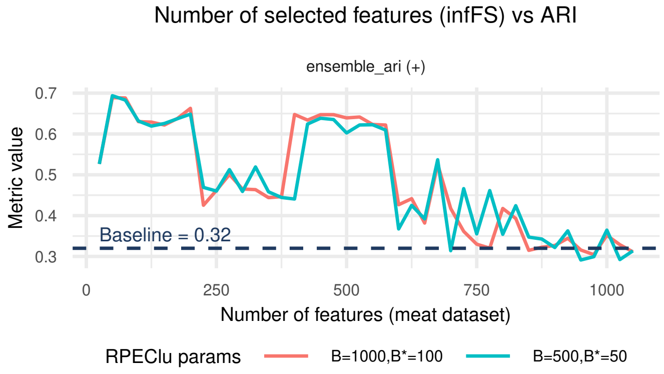

Fig. 4. Number of selected features (Inf-FS) vs ARI. Higher ARI values suggest better agreement between the clustering and ground-truth labels.

B. Experimental results of the ensemble method in benchmark datasets

After establishing the effectiveness of integrating unsupervised feature selection with ensemble clustering in the initial validation experiments, this section systematically evaluates the full ensemble clustering solution, as described in section III. This phase involves benchmarking multiple combinations of feature selection, random projection, clustering algorithms, and consensus functions across a range of high-dimensional datasets.

Table V presents the performance results of the proposed ensemble clustering methods. For each dataset, different combinations of feature selection (Inf-FS), random projection (RP), clustering algorithm, and consensus function were evaluated. The results are reported using the mean value of an external evaluation metric for each dataset (such as ARI or accuracy), along with the standard deviation over 20 runs. For comparison, reference values from previous studies are also provided.

The results demonstrate the effectiveness of integrating unsupervised feature selection (Inf-FS) and random projection (RP) within ensemble clustering frameworks. Notably, the combination of Inf-FS and RP with Gaussian Mixture Models (GMM) and consensus functions such as LCE consistently improved clustering performance in several cases, as evidenced by substantial gains in external evaluation metrics like Adjusted Rand Index (ARI) and accuracy. For instance, in the Meat and warpPIE10 datasets, the Inf-FS + RP + GMM configuration achieved ARI values of 0.69, which is significantly higher than configurations without feature selection or random projection. However, the impact of these techniques varied across datasets; in some cases, such as Lymphoma, the use of Inf-FS led to lower ARI values, underscoring the need for careful selection and tuning of ensemble components based on data characteristics. Overall, the results indicate that the proposed ensemble approach can substantially enhance clustering robustness and accuracy in high-dimensional settings.

TABLE V
PERFORMANCE OF ENSEMBLE METHODS ACROSS DATASETS.

Dataset / Ensemble	Feat.	B	Consen.	ARI / Acc ± SD
Meat (ARI=0.32 in [11])				
GMM (No RP, No UFS)	1,050	–	LCE	0.16 ± 0.02
Inf-FS + GMM (No RP)	91	10	LCE	0.48 ± 0.03
RP + GMM (No UFS)	1,050	–	LCE	0.30 ± 0.02
Inf-FS + RP + GMM	91	10	LCE	**0.69 ± 0.01**
Lung Cancer (ARI=0.31 in [55])				
GMM (No RP, No UFS)	3,313	–	Majority	**0.73 ± 0.08**
Inf-FS + GMM (No RP)	583	60	Majority	0.56 ± 0.13
RP + GMM (No UFS)	3,313	–	Majority	0.35 ± 0.05
Inf-FS + RP + GMM	583	60	Majority	0.33 ± 0.06
Lymphoma (ARI=1 in [11])				
GMM (No RP, No UFS)	4,026	–	CSPA	0.88 ± 0.16
Inf-FS + GMM (No RP)	740	80	CSPA	0.10 ± 0.05
RP + GMM (No UFS)	4,026	–	CSPA	**0.98 ± 0.02**
Inf-FS + RP + GMM	740	10	CSPA	0.29 ± 0.14
Prostate GE (ACC=0.65 in [56])				
PAM (No RP, No UFS)	5,966	–	LCE	0.58 ± 0.02
Inf-FS + PAM (No RP)	949	100	LCE	**0.78 ± 0.01**
RP + PAM (No UFS)	5,966	–	LCE	0.58 ± 0.01
Inf-FS + RP + PAM	949	100	LCE	0.69 ± 0.11
warpPIE10P (ACC=0.71 in [23])				
GMM (No RP, No UFS)	2,420	10	LCE	0.31 ± 0.03
Inf-FS + GMM (No RP)	382	10	LCE	0.61 ± 0.03
RP + GMM (No UFS)	2,420	10	LCE	0.43 ± 0.03
Inf-FS + RP + GMM	382	10	LCE	**0.72 ± 0.05**

V. CONCLUSIONS

This work demonstrates that ensemble clustering methods, when combined with unsupervised feature selection and dimensionality reduction, offer a robust solution for clustering high-dimensional datasets. The proposed framework systematically integrates diverse base clustering algorithms, feature selection (notably Inf-FS), and random projections, resulting in improved clustering accuracy, stability, and robustness compared to traditional single-method approaches. Experimental results across various benchmark datasets show that this ensemble approach can effectively mitigate common issues in high-dimensional clustering, such as noise, data sparsity, and the presence of irrelevant features, often outperforming or matching state-of-the-art results.

The key advantages of this approach include its flexibility in combining different clustering algorithms and consensus functions, its ability to adapt to a range of data types, and its demonstrated improvements in external validation metrics such as Adjusted Rand Index and accuracy.

Future work on the proposed ensemble clustering framework can focus on several promising research directions to further enhance its accuracy, adaptability, and scalability. Incorporating advanced consensus functions, such as those based on deep learning models [23], could improve performances and robustness, especially for complex and noisy datasets. Automated hyperparameter tuning and adaptive ensemble design would streamline method selection and reduce reliance on manual configuration. Integrating deep learning-based clustering approaches may allow for better modeling of nonlinear data structures, while developing scalable, distributed, or parallel algorithms can address computational bottlenecks in large-scale or streaming environments. Expanding the framework's application to diverse domains and additional real-world benchmarks will help assess its generalizability, and adding explainable AI techniques can enhance the interpretability of clustering results—crucial for scientific and medical use cases. Additionally, future research could address automatic selection of number of clusters, dynamic updating for evolving data, and more robust treatments for missing data or outliers, ensuring the framework remains effective for a wide range of high-dimensional data analysis scenario

REFERENCES

[1] R. Scitovski, K. Sabo, F. Martínez-Álvarez, and Š. Ungar, *Cluster Analysis and Applications*. Springer, 2021.

[2] J. Kleinberg, "An Impossibility Theorem for Clustering," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 15, 2002.

[3] R. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, 1961.

[4] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When Is Nearest Neighbors Meaningful?" in *International Conference on Database Theory*, pp. 217–235, 1999.

[5] X. Z. Fern and C. E. Brodley, "Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach," in *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 2003, pp. 186–193.

[6] A. Strehl and J. Ghosh, "Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, Dec. 2002.

[7] C. Domeniconi and M. Al-Razgan, ”Weighted Cluster Ensembles: Methods and Analysis,” *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 4, pp. 1–40, 2009.

[8] A. Topchy, A. K. Jain, and W. Punch, ”Combining multiple weak clusterings,” in *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM)*, Melbourne, FL, USA, 2003, pp. 331–338.

[9] A. Topchy, A. K. Jain, and W. Punch, ”Clustering ensembles: Models of consensus and weak partitions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1866–1881, 2005.

[10] X. Z. Fern and C. E. Brodley, ”Solving cluster ensemble problems by bipartite graph partitioning,” in *Proceedings of the 21st International Conference on Machine Learning (ICML)*, Banff, Alberta, Canada, 2004, pp. 36.

[11] L. Anderlucci, F. Fortunato, and A. Montanari, ”High-Dimensional Clustering via Random Projections,” *Journal of Classification*, vol. 39, no. 2, pp. 191–216, 2022.

[12] T. Boongoen and N. Iam-On, ”Cluster ensembles: A survey of approaches with recent extensions and applications,” *Computer Science Review*, vol. 28, pp. 1–25, 2018.

[13] T. Stolz, M. E. Huertas, and A. Mendoza, ”Assessment of air quality monitoring networks using an ensemble clustering method in the three major metropolitan areas of Mexico,” *Atmospheric Pollution Research*, vol. 11, no. 8, pp. 1271-1280, 2020.

[14] X. Nie, D. Qin, X. Zhou, H. Duo, Y. Hao, B. Li, and G. Liang, ”Clustering ensemble in scRNA-seq data analysis: Methods, applications and challenges,” *Computers in Biology and Medicine*, vol. 159, p. 106939, 2023.

[15] A. L. N. Fred and A. K. Jain, “Combining multiple clustering using evidence accumulation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, 2005.

[16] S. Vega-Pons and J. Ruiz-Shulcloper, “A survey of clustering ensemble algorithms,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 3, pp. 337–372, 2011.

[17] M. S. Pavithra and R. Parvathi, ”Cluster Ensemble Approach for High Dimensional Data,” *Australian Journal of Basic and Applied Sciences*, vol. 12, no. 1, pp. 45-53, Jan. 2018.

[18] X. Z. Fern and C. E. Brodley, ”Cluster Ensembles for High Dimensional Clustering: An Empirical Study,” School of Electrical and Computer Engineering, Purdue University, W. Lafayette, IN, and Electrical Engineering and Computer Science, Tufts University, Medford, MA, 2003.

[19] X. Z. Fern and W. Lin, ”Cluster ensemble selection,” *Statistical Analysis and Data Mining*, vol. 1, no. 3, pp. 128-141, 2008.

[20] J. G. Dy and C. E. Brodley, “Feature selection for unsupervised learning,” *Journal of Machine Learning Research*, vol. 5, pp. 845–889, Aug. 2004.

[21] X. He, D. Cai, and P. Niyogi, “Laplacian score for feature selection,” in *Advances in Neural Information Processing Systems*, vol. 18, 2005.

[22] G. Roffo, S. Melzi, U. Castellani, A. Vinciarelli, and M. Cristani, ”Infinite Feature Selection: A Graph-based Feature Filtering Approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[23] M. You, A. Yuan, D. He, and X. Li, “Unsupervised feature selection via neural networks and self-expression with adaptive graph constraint,” *Pattern Recognition*, vol. 135, p. 109173, Mar. 2023.

[24] S. Ayesha, M. K. Hanif, and R. Talib, ”Overview and comparative study of dimensionality reduction techniques for high dimensional data,” *Information Fusion*, vol. 59, pp. 44–58, 2020.

[25] C. Eckart and G. Young, ”The approximation of one matrix by another of lower rank,” *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936.

[26] Hotelling, Harold. “Analysis of a Complex of Statistical Variables into Principal Components.” *The Journal of Educational Psychology*, vol. 24, no. 6, 1933, pp. 417–441.

[27] W. B. Johnson and J. Lindenstrauss, ”Extensions of Lipschitz mappings into a Hilbert space,” *Contemporary Mathematics*, vol. 26, pp. 189–206, 1984.

[28] S. Vega-Pons and J. Ruiz-Shulcloper, ”Clustering Ensemble Method for Heterogeneous Partitions,” in *CIARP*, 2009, pp. 481-488.

[29] S. Dudoit and J. Fridlyand, ”Bagging to improve the accuracy of a clustering procedure,” Bioinformatics, vol. 19, no. 9, pp. 1090–1099, 2003.

[30] Z.H. Zhou, *Ensemble Methods: Foundations and Algorithms*, CRC Press, 2012.

[31] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, 2002, pp. 849–856.

[32] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, 1996, pp. 226–231.

[33] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “OPTICS: Ordering points to identify the clustering structure,” in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, 1999, pp. 49–60.

[34] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum Press, 1981.

[35] V. Berikov, ”Weighted ensemble of algorithms for complex data clustering,” *Pattern Recognition Letters*, vol. 38, pp. 99-106, 2014.

[36] K. Golalipour, E. Akbari, S. S. Hamidi, M. Lee, and R. Enayatifar, “From clustering to clustering ensemble selection: A review,” *Engineering Applications of Artificial Intelligence*, vol. 104, p. 104388, 2021.

[37] M. C. Naldi, A. C.P.L. de Carvalho, and R. J.G.B. Campello, “Cluster ensemble selection based on relative validity indexes,” *Data Mining and Knowledge Discovery*, vol. 27, no. 2, pp. 259–289, 2013.

[38] E. Akbari, H. M. Dahlan, R. Ibrahim, and H. Alizadeh, “Hierarchical cluster ensemble selection,” *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 146–156, 2015.

[39] S. O. Abbasi, S. Nejatian, H. Parvin, V. Rezaie, and K. Bagherifard, “Clustering ensemble selection considering quality and diversity,” *Artificial Intelligence Review*, vol. 52, no. 2, pp. 1311–1340, 2019.

[40] H. W. Kuhn, ”The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[41] H. G. Ayad and M. S. Kamel, ”Cumulative voting consensus method for partitions with a variable number of clusters,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 160–173, 2008.

[42] A. P. Topchy, M. H. C. Law, A. K. Jain, and A. L. N. Fred, ”Analysis of consensus partition in cluster ensemble,” in *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM)*, 2004, pp. 225–232.

[43] J. M. Luna-Romera, M. Martínez-Ballesteros, J. García-Gutiérrez, and J. C. Riquelme, “External clustering validity index based on chi-squared statistical test,” *Information Sciences*, vol. 487, pp. 1–17, 2019.

[44] D. S. Chiu and A. Talhouk, ”diceR: an R package for class discovery using an ensemble driven approach,” BMC Bioinformatics, vol. 19, no. 1, p. 11, Feb. 2018.

[45] D. Chiu, A. Talhouk, and J. Liu, ”Package ’diceR’,” CRAN, Feb. 2025. [Online]. Available: https://CRAN.R-project.org/package=diceR

[46] A. C. Rodriguez Bajo, “diceRplus,” GitHub repository, 2025. [Online]. Available: https://github.com/antoniocarlosrodriguezbajo/diceRplus

[47] F. Abedinzadeh Torghabeh, Y. Modaresnia, and S. A. Hosseini, ”Auto-UFSTool: An Automatic Unsupervised Feature Selection Toolbox for MATLAB,” *Journal of AI and Data Mining*, 2023.

[48] H. Bengtsson, ”R.matlab: Read and Write MAT Files and Call MATLAB from Within R,” R package version 3.7.0, Jan. 2025. [Online]. Available: https://github.com/HenrikBengtsson/R.matlab

[49] A. C. Rodriguez-Bajo, “diceRplus,” GitHub repository, 2025. [Online]. Available: https://github.com/antoniocarlosrodriguezbajo/diceRplus

[50] J. McElhinney, G. Downey, and T. Fearn, ”Chemometric processing of visible and near infrared reflectance spectra for species identification in selected raw homogenised meats,” *Journal of Near Infrared Spectroscopy*, vol. 7, pp. 145–154, 1999.

[51] A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, et al., ”Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinomas sub-classes,” *Proc. Natl. Acad. Sci.*, vol. 98, no. 24, pp. 13790–13795, 2001.

[52] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, et al., ”Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling,” *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.

[53] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, et al., ”Gene expression correlates of clinical prostate cancer behavior,” *Cancer Cell*, vol. 1, no. 2, pp. 203–209, 2002.

[54] J. Li, K. Wang, et al., ”scikit-feature: A feature selection repository.” Dataset available at: https://jundongl.github.io/scikit-feature/datasets.html

[55] S. Monti, P. Tamayo, J. P. Mesirov, T. R. Golub, ”Consensus clustering: A resampling-based method for class discovery and visualization of gene

expression microarray data," *Machine Learning*, vol. 52, no. 1–2, pp. 91–118, 2003.

[56] J. Guo, W. Zhu, "Dependence Guided Unsupervised Feature Selection," in *Proc. 32nd AAAI Conf. Artificial Intelligence (AAAI-18)*, vol. 32, pp. 2232–2239, 2018.

Antonio Carlos Rodríguez Bajo received his B.S. degree in Applied Data Science in 2023 from the Universitat Oberta de Catalunya, where he was recognized with the Bachelor's Degree Extraordinary Award. He is currently pursuing his Master's Degree in Artificial Intelligence Research at the Universidad Internacional Menéndez Pelayo, in collaboration with the Spanish Association for Artificial Intelligence (AEPIA).