CTU Journal of Innovation and Sustainable Development
ISSN 2588-1418 | e-ISSN 2815-6412

DOI:10.22144/ctujoisd.2025.055

Component-based ensemble cluster analysis

Al Maruf Hassan[1], Huu-Hoa Nguyen[2*], Md. Maruf Hassan[3], Abdul Kadar Muhammad Masum[3], and Dewan Md. Farid[3]

[1]*Department of Electrical and Computer Engineering, North South University, Bangladesh*

[2]*College of Information and Communication Technology, Can Tho University, Viet Nam*

[3]*Department of Computer Science and Engineering, Southeast University, Bangladesh*

**Corresponding author (nhhoa@ctu.edu.vn)*

Article info.

Received 14 Jul 2025
Revised 16 Aug 2025
Accepted 8 Oct 2025

Keywords

Clustering, component-based clustering, ensemble clustering

ABSTRACT

Ensemble clustering leverages multiple methods to identify diverse patterns and, instead of depending on a singular approach, generates a more dependable and accurate clustering solution. This methodology mitigates bias and noise in intricate, high-dimensional data, allowing the grouping of biological and genomic big data. Component-based ensemble clustering divides data into subsets, applies several algorithms, and then aggregates the outcomes to increase performance. This method analyzes each data subset independently, facilitating the recognition of various patterns while minimizing noise and bias. This paper proposes two novel clustering methods that integrate multiple algorithms, including Agglomerative Hierarchical Clustering (AHC), K-Means Clustering, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), Ordering Points to Identify the Clustering Structure (OPTICS), Improved Density-Based Spatial Clustering of Applications with Noise (IDBSCAN), and Density-Based Spatial Clustering of Applications with Noise Plus Plus (DBSCAN++). The second method, termed Ensemble Clustering with Each Subset (ECES), employs both 'with-replacement' and 'without-replacement' techniques to increase variety, minimize redundancy, and improve generalization. The key distinction resides in the ensemble step of the second strategy, which divides datasets into equal subsets to ensure fairness and comparability. This ensures fairness, comparability, and controlled diversity within the ensemble, reducing bias, redundancy, and overlap.

1. INTRODUCTION

Clustering or data segmentation in unsupervised learning in machine learning and data mining research is the process of grouping the data instances into clusters, so that instances within a cluster have high similarity in comparison to one another but are very dissimilar to instances in other clusters. Similarities and dissimilarities of instances are based on the attribute values described in the instances. Cluster analysis is the process of partitioning a set of data instances into subsets. Each subset is a *cluster*, such that instances in a cluster are similar to one another, yet dissimilar to instances in other clusters. The set of clusters resulting from a cluster analysis can be referred to as a *clustering*. It can lead to the discovery of previously unknown groups within the data. Different clustering methods may generate different clustering of the same

dataset. Cluster analysis has been widely used in many applications such as business intelligence, Web search, biology, security, anthropology, pattern recognition, and image processing. Clustering is sometimes called *automatic classification*. It is also called *data segmentation* in some applications because clustering partitions large data sets into groups according to their *similarity* (Farid et al., 2019).

Clustering is a form of learning by observation. Data clustering has recently become a highly active research topic because assigning class labels to numerous instances can be a very costly process. The goal of clustering is to determine the intrinsic grouping of a set of unlabeled data. It is the process of grouping the instances into clusters (or classes). Dissimilarities are assessed based on the attribute values describing the instances. Also, a cluster usually should consist of a group of instances that are similar to one another and are dissimilar to instances in other clusters. There are many typical requirements of clustering in machine learning, e.g., clustering big data, constraint-based clustering, dealing with noisy data, etc. Clustering many data instances is a very costly process. Most of the existing clustering algorithms work well on small data sets containing fewer than several hundred data instances with few attributes; however, a large data set may contain millions of data instances with numerous attributes (Farid et al., 2019).

Component-Based Ensemble Clustering is an extension of ensemble clustering, where instead of only combining whole partitions from multiple clustering solutions, it exploits the *substructures* (components) hidden inside those clusters (Zheng et al., 2025). Component-Based Ensemble Clustering fills the gap by providing a more stable, robust, and fine-grained clustering approach that can handle noisy, high-dimensional, and heterogeneous real-world data in significant areas of *healthcare, social networks, bioinformatics, and business intelligence*, where traditional clustering and even standard ensemble clustering often fail (Ren et al., 2025; Yang et al., 2025).

In this paper, we have presented two novel component-based ensemble clustering methods named Independent Heterogeneous Ensemble Clustering (IHEC) and Ensemble Clustering with Each Subset (ECES-with and without replacement). The main contributions of this paper are summarized as follows:

– We have proposed two algorithms named, respectively, IHEC and ECES-with and without replacement techniques employing Agglomerative Hierarchical Clustering (AHC), K-Means Clustering, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), Ordering Points To Identify the Clustering Structure (OPTICS), Improved Density-Based Spatial Clustering of Applications with Noise (IDBSCAN), Density-Based Spatial Clustering of Applications with Noise Plus Plus (DBSCAN++)) clustering algorithms on 10 benchmark datasets.

– The proposed clustering methods aim to compare the performance of different clustering algorithms in different scenarios and perform disjoint and non-disjoint subsets to reduce the redundancy, multicollinearity, overfitting, and curse of dimensionality.

– We have evaluated the performance of Agglomerative Hierarchical Clustering (AHC), K-Means Clustering, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), Ordering Points To Identify the Clustering Structure (OPTICS), Improved Density-Based Spatial Clustering of Applications with Noise (IDBSCAN), Density-Based Spatial Clustering of Applications with Noise Plus Plus (DBSCAN++)) clustering algorithms through the two proposed approaches using two techniques (i.e., disjoint and non-disjoint subsets) and exploring the patterns and behaviors of the employed clustering algorithm in different dimensions.

The rest of the paper is structured as follows: Section 2 discusses the literature review. Section 3 discusses ensemble clustering and proposes clustering algorithms. Then, experimental analysis and dataset description are shown and discussed in the Section 4. Conclusion and future works are presented in Section 5.

2. LITERATURE REVIEW

Hong et al. (2019) proposed a Gaussian mixture model that captures feature-specific influences on mixture components, enabling a new component-level feature saliency measure. Using Markov Chain Monte Carlo for estimation, their method outperforms traditional feature saliency approaches in clustering accuracy and parameter estimation on synthetic data. To address the challenge of choosing the best clustering algorithm for gene expression data, Vukicevic et al. (2016) developed an advanced meta-learning framework. It enhances earlier

models by enlarging the pool of algorithms, broadening the dataset descriptors (meta-features), and applying cutting-edge techniques for feature selection and model tuning (Tian & Zhang, 2025). This method is tested extensively—across 504 algorithms and 30 datasets—and proved highly effective in predicting which algorithms would perform best for specific data scenarios (Liu et al., 2021).

Li (2010) introduced two new methods— Multi Optimisation Consensus Clustering (MOCC) and K-Ants Consensus Clustering (KACC)—to boost ensemble clustering performance that leverages heuristic optimization strategies (Simulated Annealing and Ant Colony Optimisation) for better consensus clustering. These approaches showed superior accuracy compared to existing techniques, with results and in-depth evaluations presented in his research. Chen et al. (2025) presented contrastive ensemble clustering (CEC), a novel ensemble clustering approach that leverages latent representation learning and contrastive regularisation to extract meaningful patterns from noisy data. By combining a consensus model with a locality-preserving contrastive component, CEC delivers superior clustering performance and pioneers the use of contrastive learning in ensemble clustering (Zhou et al., 2025). Zhang et al. (2025) introduced Structured Bipartite Graph Learning (SBGL), which enhances ensemble clustering by constructing bipartite graphs from sample-cluster similarity matrices of base clustering. These graphs are projected into sample-latent-cluster graphs (Zhan et al., 2025), which are then combined into a unified bipartite graph with a clear cluster structure. The final clustering is extracted from this graph. SBGL accommodates varying numbers of clusters across base clustering, contributing to improved overall performance.

Xu et al. (2025) proposed Sparse Dual-Weighting Ensemble Clustering (SDWEC), which improves clustering by weighting base clusterings and their clusters while enforcing sparsity to select informative components. It directly learns cluster indicators, reduces information loss, and uses an efficient convergent linear-time optimization algorithm. Mahmud et al. (2025) proposed an ensemble clustering method for large-scale data using the RSPCA framework, which partitions data into random, distribution-preserving blocks. A subset of these blocks is clustered individually, and the results are aggregated to approximate the full data clustering. The process supports incremental updates for greater robustness. The I-niceDP algorithm estimates the number of clusters, while the k-means refines the centroids. Spectral and correlation clustering are consensus functions that handle complex cluster patterns (Shang, 2025; Wei, 2025).

In our literature review, we did a rigorous exploration, and no direct research work was found on our research, Component-Based Ensemble Clustering. We cannot provide a direct comparison with other state-of-the-art approaches for the reasons mentioned above.

3. ENSEMBLE CLUSTERING

Ensemble clustering is an approach that combines multiple clustering algorithms to create a robust and effective clustering solution, usually producing superior results compared to individual methods (Li, 2025; Yu, 2025). The primary objective of ensemble clustering is to consolidate the outcomes of various clustering methods into a single and more accurate clustering result. Let $X = x_1, x_2, \ldots, x_N$ denotes an unlabeled dataset consisting of N instances. The task of clustering is to partition X into k clusters $C_1, C_2, \ldots, C_k$, satisfying the conditions:

$$C_i \neq \emptyset, for\ i = 1,2,\ldots,k.\ where \bigcup_{i=1}^{k} C_i = X;$$

$$C_i \cap C_j = \emptyset, and\ for\ i \neq j, \quad i,j = 1,2,\ldots,k. \quad (1)$$

Given multiple clustering results obtained from different clustering algorithms or different parameter settings, an ensemble clustering approach aims to integrate these results into a single consensus clustering that achieves higher accuracy and robustness. Formally, suppose we have a set of M clustering algorithms.

$\mathcal{A} = A_1, A_2, \ldots, A_M$ applied to dataset X. Each algorithm A_m produces a partition. The ensemble clustering problem can be mathematically represented as a function $\mathcal{F}$ that maps the set of partitions into a final consensus partition $\prod *$.

3.1. Component-Based Ensemble

Component-based ensemble clustering enhances unsupervised learning on complex, high-dimensional data by pre-processing and grouping relevant instances, clustering each group independently, and merging results via ensemble methods. This modular strategy improves accuracy, scalability, and robustness. Consider an unlabeled data set, where a set of features characterizes each instance. Ensemble clustering based on components

divides the data into separate or overlapping subsets (components) to leverage structural differences and mitigate the impact of dimensionality and noise.

Formally, let it be partitioned into distinct components:

$$X = X_1, X_2, \dots, X_S, where \bigcup_{s=1}^{S} X_s = \text{X and } X_i \cap X_j = \emptyset, \forall i \neq j. \quad (2)$$

In set theory, two components (or sets) are said to be non-disjoint if they share at least one common element. Formally, for two components A and B, this condition is expressed as

$$A \cap B \neq \emptyset. (3)$$

In general, a partition of a set requires the components (subsets) to be mutually disjoint. However, when the components are non-disjoint, the partition condition is relaxed, allowing overlaps among components.

Formally, let a set X be covered by a family of components $C = \{C_1, C_2, \dots, C_k\}$ such that

$$\bigcup_{i=1}^{k} C_i = X. \quad (4)$$

If there exist indices $i \neq j$ such that

$$Ci \cap C_j \neq \emptyset, \quad (5)$$

then the family $\mathcal{C}$ constitutes a cover of X by non-disjoint components.

Each component is clustered independently clustered, using clustering algorithms. Denote by the clustering outcome the i-th clustering algorithm applied to the component, given as:

$$C^{(q)}(X_s) = C^{(q)}s, 1, C^{(q)}s, 2, \dots, C^{(q)}_{s,K_s^{(q)}}, (6)$$

Where represents the number of clusters produced by the i-th clustering algorithm applied to the component. The ensemble clustering problem aims to integrate these individual clustering outcomes into a single consensus clustering, represented as:

$$C^* = Consensus\left(C^{(q)}(Xs)._{q=1,\dots,Q,s=1,\dots,s}\right). (7)$$

The consensus clustering maximises a clustering validity measure, such as cluster compactness or separation, subject to a constraint on minimising disagreement between component clusters (Hu & Rezaeipanah, 2025). Formally, the consensus clustering objective can be expressed as follows.

$$C^* = \arg\max_{C} \sum_{s=1}^{S} \sum_{q=1}^{Q} Sim\left(C, C^{(q)}(X_s)\right), \quad (8)$$

Where $Sim(i,j)$ denotes a measure of similarity between the outcomes of the cluster.

3.2. Proposed Clustering Methods

We have taken a dataset as input, D. The dataset has a set of features X_i, where N is the number of attributes of a dataset. We employed six existing clustering algorithms: (1) Agglomerative Hierarchical Clustering (AHC), (2) K-Means Clustering, (3) Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), (4) Ordering Points To Identify the Clustering Structure (OPTICS), (5) Improved Density- Based Spatial Clustering of Applications with Noise (IDBSCAN), (6) Density-Based Spatial Clustering of Applications with Noise Plus Plus (DBSCAN++)) on 10 datasets.

In the first concept, named Independent Heterogeneous Ensemble Clustering (IHEC), we directly applied six existing clustering algorithms: (1) Agglomerative Hierarchical Clustering (AHC), (2) K-Means Clustering, (3) Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), (4) Ordering Points To Identify the Clustering Structure (OPTICS), (5) Improved Density-Based Spatial Clustering of Applications with Noise (IDBSCAN), (6) Density-Based Spatial Clustering of Applications with Noise Plus Plus (DBSCAN++)) on 10 datasets and analysis the performances of the each clustering algorithms. The IHEC is a baseline concept in ensemble cluster analysis, and it does not involve any part of the concept of component-based ensemble clustering.

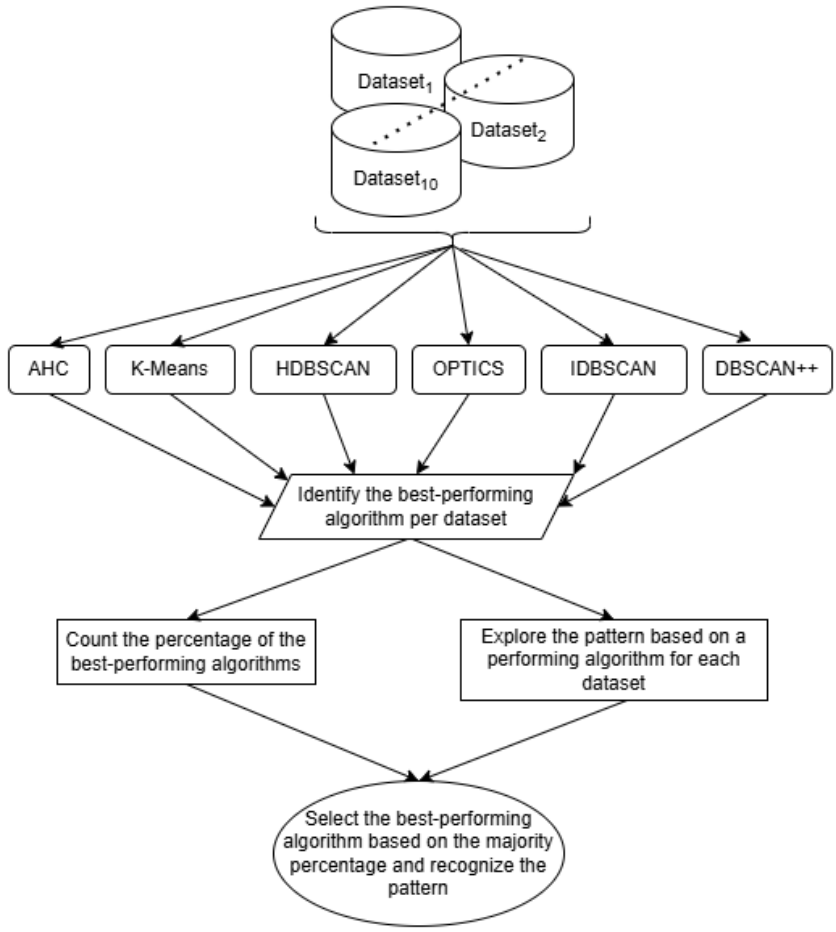

Figure 1. Independent Heterogeneous Ensemble Clustering (IHEC)

In the second concept, named Ensemble Clustering with Each Subset (ECES), we divided the dataset into an equal number of subsets (m, k) as per the employed six clustering algorithms, according to the technique of with and without replacement. Then, we have identified the best-performing algorithm for individual datasets according to the performance measure metrics in the clustering problem called cluster compactness. We have counted the percentage of the best-performing algorithms and explored the patterns based on the best-performing algorithm for each dataset, based on the minimum compactness score of the cluster. We have selected the majority percentage of the best-performing algorithm with different techniques (i.e., independent, with and without replacement) and recognised the patterns that indicate which algorithm with a specific technique performs well in which dataset type.

Implementation of the proposed clustering algorithms 1 and 2, named, respectively, Independent Hetero- geneous Ensemble Clustering (IHEC) and Ensemble Clustering with Each Subset (ECES), can be accessed from the GitHub page (https://github.com/marufgreat/Component-BasedEnsembleClustering.git).

Algorithm 1: Independent Heterogeneous Ensemble Clustering (IHEC)

Require: Dataset $X = \{x_1, x_2, \ldots, x_n\}$

Ensure: Best clustering result C^* based on minimum cluster compactness (comp.)

1: Initialize clustering algorithms $\mathcal{A}$ = {AHC, KMeans, OPTICS, HDBSCAN, IDBSCAN, DBSCAN++}

2: Initialize an empty list of clusterings $\mathcal{C} \leftarrow [\,]$

3: Initialize an empty list of compactness scores $\mathcal{S} \leftarrow [\,]$

4: for each algorithm $A\_i \in \mathcal{A}$ do

5: Apply A_i to the full dataset X to obtain clustering Ci

6: Compute compactness score $s_i \leftarrow \text{ComputeCompactness}(X, C_i)$

7: Append C_i to $\mathcal{C}$

8: Append s_i to $\mathcal{S}$

9: end for

10: $C^* \leftarrow \mathcal{C}[\arg\min(\mathcal{S})]$ ▷ Select clustering with minimum compactness

11: return C^*

12: function COMPUTEDCOMPACTNESS(D, C)

13: $k \leftarrow$ number of unique clusters in C, excluding noise (-1 if present)

14: TotalCompactness ← 0, ValidClusters ← 0

15: for each cluster label c ∈ C do

16: if $|C_c| > 1$ and $c \neq -1$ then

17: P ← all pairs $x_i, x_j \in C_c$

18: DistSum $\leftarrow \sum_{x_i, x_j \in P} \left|\left|x_i - x_j\right|\right|$

19: PairCount $\leftarrow \binom{|C_c|}{2}$

20: $\text{Compactness}_c \leftarrow \frac{DistSum}{PairCount}$

21: TotalCompactness ← TotalCompactness + Compactness_c

22: ValidClusters ← ValidClusters + 1

23: end if

24: end for

25: if ValidClusters = 0 then

26: return ∞ ▷ All clusters are noise or singletons

27: end if

28: return $\frac{TotalCompactness}{ValidClusters}$

29: end function

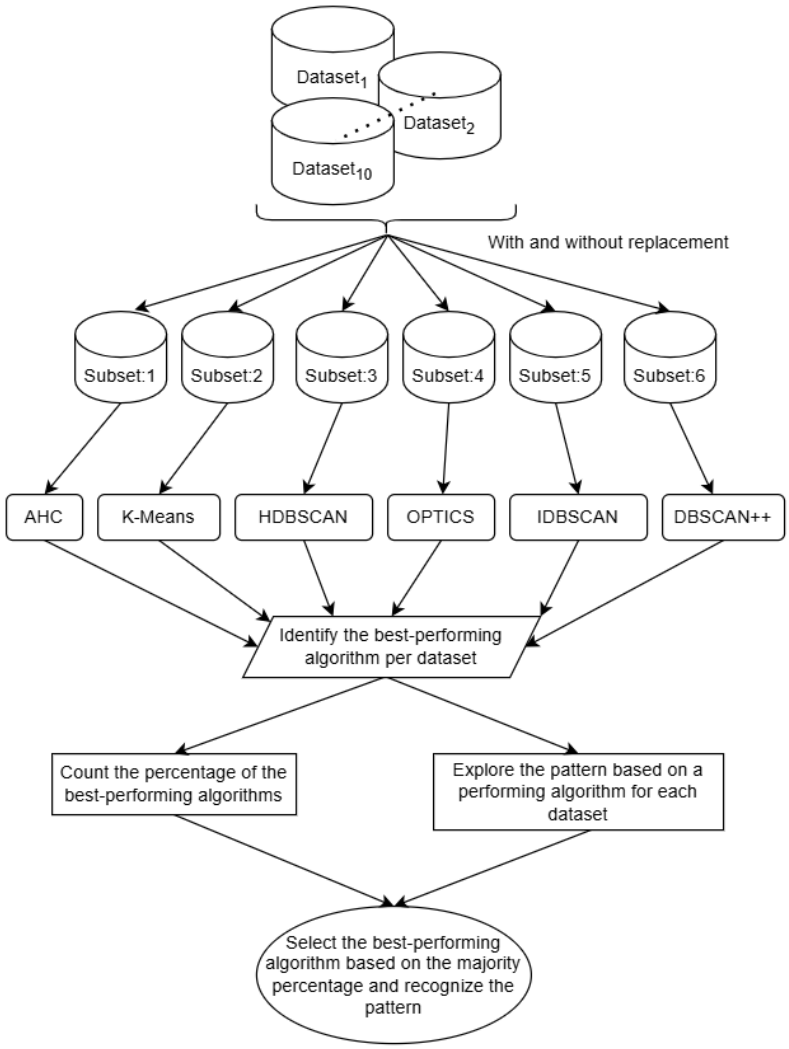

Figure 2. Ensemble Clustering with Each Subset (ECES)

Algorithm 2: Ensemble Clustering with Each Subset (ECES)

1: Input: Dataset $X = \{x_1, x_2, \dots, x_n\}$

2: Output: Final clustering labels L_{final} with minimum cluster compactness(comp.)

3: Divide X into m disjoint (without replacement) subsets $X_1, X_2, \dots, X_m$ ▷ One per algorithm

4: Divide X into k non-disjoint (with replacement) subsets $X_1, X_2, \dots, X_k$ ▷ One per algorithm

5: Define clustering algorithms $\mathcal{A}$ = {AHC, K-Means, OPTICS, HDBSCAN, IDBSCAN, DBSCAN++}

6: for each subset X_i in $\{X_1, \dots, X_m\}$ do

7: for each algorithm A_j in $\mathcal{A}$ do

8: Run A_j on X_i to obtain clustering C_{ij}

9: Compute compactness CP_{ij} = Compactness(C_{ij})

10: end for

11: Select C_i^{best} argmin_j CP_{ij}

12: Assign L_i = Labels (C_i^{best})

13: end for

14: for each subset X_l in $\{X_1 \dots, X_k\}$ do

15: for each algorithm A_z in $\mathcal{A}$ do

16: Run A_z on X_l to obtain clustering Clz

17: Compute compactness CP_{lz} = Compactness (C_{lz})

18: end for

19: Select C_l^{best} = arg min$_z$ CP_{lz}

20: Assign L_l = Labels (C_l^{best})

21: end for

22: Concatenate all labels L_l to form L_{final}

23: return L_{final}

24: function COMPACTNESS(C)

25: Let $C = \{C_1 \dots, C_k\}$ be the set of clusters

26: $total \leftarrow 0$; $count \leftarrow 0$

27: for each cluster C_l in C do

28: if $|C_l| > 1$ then

29: Compute average intra-cluster distance:

$$\mathrm{d}(C_l) = \frac{1}{|C_l|(|C_L| - 1)} \sum_{x,y \in C_l, x \neq y} ||x - y||$$

30: total ← total + d(C_l)

31: count ← count + 1

32: end if

33: end for

34: if count = 0 then

35: return ∞ d ▷All clusters are noise or singletons

36: else

37: return $total/count$

38: end if

39: end function

4. EXPERIMENTAL ANALYSIS

We have used 10 benchmark datasets in this experiment, and the dataset details are shown in Table 1.

Let $X = \{x_1, x_2, \dots, x_N\}$ be a set of N data instances in Rm, and let this set be partitioned into k

clusters $\{C_1, C_2, \dots, C_k\}$. Each cluster C_l contains n_l instances such that $\sum_{l=1}^{k} n_l = N$.

The compactness of the clustering, denoted as CP, is defined as the average pairwise intra-cluster distance, and is given by:

$$CP = \frac{1}{N}\sum_{l=1}^{k} n_l \left(\frac{1}{n_l(n_l-1)/2}\sum_{x_i,x_j\in C_l, i<j} d(x_i,x_j)\right) (9)$$

where $d(x_i, x_j)$ is the distance between instances xi and xj within the same cluster C_l . Typically, the Euclidean distance is used:

$$d(x_i,x_j) = \left|\left|x_i - x_j\right|\right|_2 = \sqrt{\sum_{r=1}^{m}(x_{ir} - x_{jr})^2} (10)$$

A lower value of CP indicates that instances within the same cluster are more tightly packed (i.e., more similar), suggesting a better clustering result in terms of compactness. Thus, minimizing CP is often desirable when evaluating or optimizing clustering algorithms.

1. Cluster Compactness = 0 (zero)

– This implies that all instances in every cluster are identical or coincide at the same point: $d(x_i, x_j) = 0$ for all $x_i, x_j \in C_l$. – This represents a perfect compactness scenario, where intra-cluster distances are minimized.

2. Cluster Compactness = ∞ (infinity)

– This indicates that one or more clusters contain either: • Singleton clusters (i.e., $n_l = 1$, making the denominator zero and the term undefined), or • Instances that are extremely far apart (i.e., $d(x_i, x_j) \to \infty$).

– Practically, such a value reflects poorly formed clusters or anomalies in the clustering process, such as noise or misconfigured parameters.

Table 1. Dataset Description

No.	Datasets	No.of Features	Feature Types	Instances	No.of classes (labels)
1.	Breast cancer	9	Nominal	286	2
2.	Wine	13	Numerical	178	3
3.	Diabetes	8	Numerical	768	2
4.	Glass	9	Numerical	214	7
5.	Seeds	7	Numerical	210	3
6.	Magic	10	Numerical	19020	2
7.	Vote	16	Nominal	435	2
8.	Fertility	8	Numerical	100	2
9.	Tic-Tac-Toe	9	Nominal	958	2
10.	Lymphography	18	Numerical	148	4

4.1. Experimental setup

We take Google Colab 5, a platform hosted in the cloud for coding using Python 3.x (version 3.13.5). We consider TensorFlow 6 (version 2.19), an open-source library for running machine learning algorithms. We also consider Scikit Learn (version 1.7.0) 7 for applying traditional clustering algorithms (i.e., Agglomerative Hierarchical Clustering (AHC), K-Means Clustering, HDBSCAN, OPTICS, IDBSCAN, DBSCAN++, etc.). We take the NumPy and Pandas frameworks, that utilize straightforward techniques for handling and manipulating scientific data. We use the Matplotlib framework for plotting, subplots, and constructing images.

4.2. Result and discussion

Tables 2, 3, and 4 show the comparison results of cluster compactness (CCp) of proposed approaches, respectively, Independent Heterogeneous Ensemble Clustering (IHEC), Ensemble Clustering with Each Subset (ECES-Without Replacement), and Ensemble Clustering with Each Subset (ECES-With Replacement). FIGURE 3, 4, and 5 illustrate the comparison of experiment cluster compactness (CCp) result and behaviour of the proposed approaches, respectively, IHEC, ECES-without replacement, and ECES-with replacement, with the x-axis denoting the value of CCp that we take from our experiment and the y-axis denoting each benchmark dataset that we employ in our experiment. In Table 2 and Figure 3, the OPTICS clustering algorithm performs outstandingly in our novel IHEC technique. Its remarkable performance is observed through all the datasets taken in our experiment, except the magic dataset. It shows about 90% of the total experimental datasets.

In Table 3 and Figure 5, the OPTICS clustering algorithm also shows outstanding performance in our novel technique of ECES without replacement.

Its impressive performance is observed across all the datasets in our experiment, except the Tic-Tac-Toe dataset. It also shows that across 90% of the total experimental datasets, the same as another novel technique of IHEC that was previously discussed.

In Table 4 and Figure 4, we observed from the experiment that a single clustering algorithm does not show outstanding performance for all the datasets, as previously mentioned, novel techniques of IHEC and ECES-without replacement. The OPTICS clustering algorithm performs well with our novel technique of ECES-with replacement in the following datasets (i.e., Magic, Vote, Fertility, and Tic-Tac-Toe) compared to the other clustering algorithms (i.e., AHC, K-Means, HDBSCAN, IDBSCAN, DBSCAN++). It shows about 40% of the total experimental datasets.

We get the best results from the Agglomerative Hi-Hierarchical Clustering (AHC) clustering algorithm to employ our novel ECES-with-replacement technique. The AHC clustering algorithm performs well in the following datasets (i.e., Breast cancer, Wine, Diabetes, Glass, and Seeds) compared to the other clustering algorithms (i.e., K-Means, OPTICS, HDBSCAN, IDB-SCAN, DBSCAN++). It shows approx. 50% of the total experimental datasets. Eventually, in 10% cases, the lymphography dataset with the K-Means clustering algorithm performs well compared to the other clustering algorithms in the technique of ECES-with replacement.

We found a significant pattern in our experiments. The OPTICS clustering algorithm performs poorly only for the high-dimensional dataset (i.e., magic (19,020, 10)) in our IHEC technique compared to our other experimental datasets. Here, the magic dataset has the highest number of samples (19,020) compared to all our experimental datasets. The DBSCAN++ clustering algorithm performs very well, especially in this case, compared to other clustering algorithms. The OPTICS clustering algorithm with ECES-with replacement technique performs well for the high-dimensional datasets (i.e., magic (19,020, 10), tic-tac-toe (958, 9)) compared to our other experimental datasets.

The magic and tic-tac-toe datasets hold the highest number of instances or samples, respectively, 19,020 and 958, compared to other total experimented datasets in our experiment. On the contrary, the K- Means and AHC clustering algorithms perform well on both a high number of features and a few instances or samples compared to the number of features ratio of the datasets, respectively lymphography (148, 18), and Wine (178, 13), Diabetes (768, 8) as well as compared to other experimented datasets in our experiment.

Table 2. Comparison of Cluster Compactness of IHEC algorithm

Datasets	AHC	K-Means	HDBSCAN	OPTICS	IDBSCAN	DBSCAN++
Breast cancer	3.6712	3.6426	2.4079	**1.6306**	2.6048	4.2072
Wine	3.9059	3.5629	2.8102	**2.0622**	2.9904	4.9521
Diabetes	3.3885	3.3872	2.2250	**1.4020**	2.1137	2.3605
Glass	2.8834	3.3914	1.5025	**0.9479**	1.4596	3.2385
Seeds	1.7208	1.8536	1.8461	**0.8929**	1.2800	1.8821
Magic	1.2322	1.2376	0.8142	0.4005	0.6372	**0.3087**
Vote	4.7213	4.5993	1.3653	**0.6176**	2.9975	5.3450
Fertility	3.6759	3.7167	2.4659	**2.0340**	3.0673	4.3522
Tic-Tac-Toe	3.8482	3.8294	4.1109	**2.4818**	**2.4818**	3.8827
lymphography	5.3664	5.2568	3.0397	**2.5452**	5.0683	5.4693

Table 3. Comparison of Cluster Compactness of ECES - without replacement

Datasets	AHC	K-Means	HDBSCAN	OPTICS	IDBSCAN	DBSCAN++
Breast cancer	3.4596	3.4028	2.6645	**1.6813**	3.2547	4.2072
Wine	3.5234	3.4092	2.7963	**2.7606**	3.9257	4.9521
Diabetes	3.2507	3.0315	3.0027	**1.6179**	2.8323	3.7592
Glass	2.4390	2.3263	2.2847	**0.9691**	2.0835	3.2385
Seeds	1.8873	1.8397	1.8500	**1.4957**	1.6471	1.8822
Magic	inf.	inf.	inf.	**3.2880**	inf.	inf.
Vote	4.1106	4.0247	2.0820	**0.8304**	3.2582	5.3450
Fertility	3.6788	3.6084	2.5539	**2.0940**	3.8206	4.3522
Tic-Tac-Toe	3.9042	**3.8632**	4.1720	4.0155	4.0155	4.1697
lymphography	4.8448	4.9193	3.3583	**2.5452**	5.2418	5.4693

Table 4. Comparison of Cluster Compactness of ECES - with replacement

Datasets	AHC	K-Means	HDBSCAN	OPTICS	IDBSCAN	DBSCAN++
Breast cancer	**0.3981**	3.6131	2.4079	2.3963	0.4657	inf.
Wine	**0.9261**	3.4990	3.5475	2.8221	inf.	inf.
Diabetes	**0.7529**	3.5023	2.3165	1.7558	inf.	inf.
Glass	**0.3978**	3.7015	inf.	0.8400	inf.	inf.
Seeds	**0.9301**	1.9338	2.3019	1.3891	inf.	inf.
Magic	3.7773	3.6867	3.6230	**2.9219**	inf.	inf.
Vote	0.0000	4.1355	3.5012	**1.8972**	inf.	inf.
Fertility	0.0000	3.1890	inf.	**2.7153**	inf.	inf.
Tic-Tac-Toe	0.0000	3.8143	3.0983	**2.9225**	inf.	inf.
lymphography	0.0000	**2.6956**	inf.	5.7306	inf.	inf.

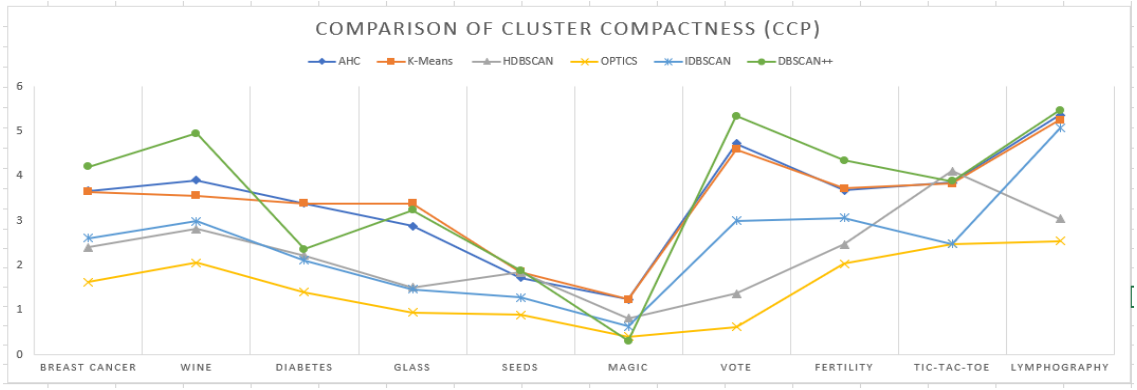

Figure 3. Independent Heterogeneous Ensemble Clustering (IHEC)

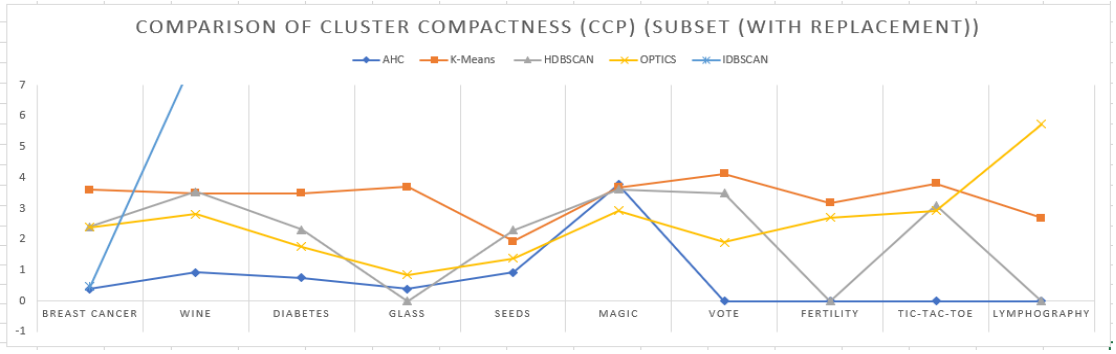

Figure 4. Ensemble Clustering with Each Subset (ECES-with replacement)

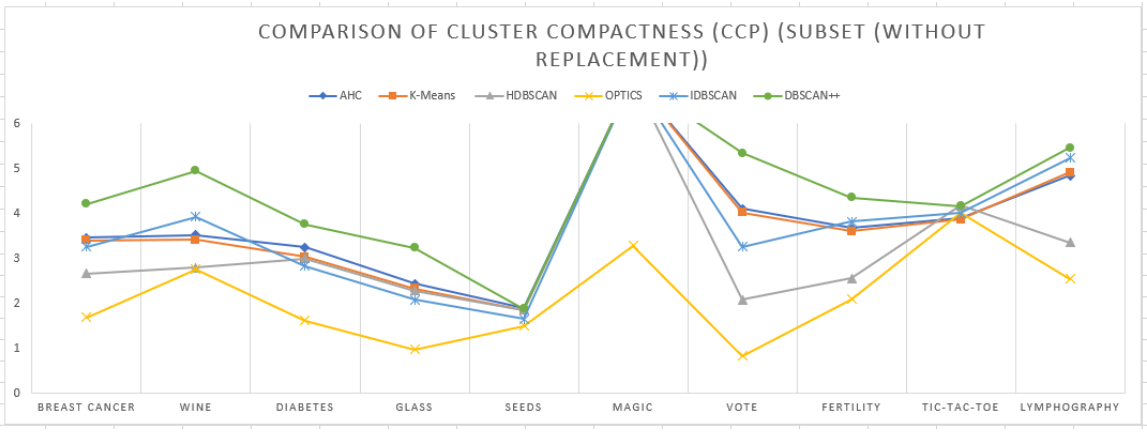

Figure 5. Ensemble Clustering with Each Subset (ECES-without replacement)

5. CONCLUSION AND FUTURE WORK

This paper presents two clustering techniques amalgamated with six clustering algorithms: AHC, K-Means, HDBSCAN, OPTICS, IDBSCAN, and DBSCAN++. We have introduced the with-replacement and without- replacement techniques in the second approach, Ensemble Clustering with Each Subset (ECES), to explore the diversity of the datasets and ensure the diversity, redundancy, and generalisation capability of the proposed clustering techniques. The key difference between the first and second approaches is the equal number of subsets employing clustering algorithms called ensemble clustering. We do the ensemble clustering through the second approach to ensure fairness and comparability, reduce bias, control ensemble diversity, reduce redundancy and overlap, improve generalisation assessment, simplify evaluation, and fusion. In the first approach, we directly employed each clustering algorithm on the datasets without using any ensemble technique to explore the patterns and behavior of each clustering algorithm, like AHC, K-Means, HDBSCAN, OPTICS, IDBSCAN, and DBSCAN++. In the future, we will apply subspace search methods (i.e., bottom- up approaches like CLIQUE, top-down approaches, and metaheuristic-based approaches) with component clustering. The subspace search method broadly refers to optimization or learning approaches that, rather than exploring the entire (often high-dimensional) solution space, confine the search to a lower-dimensional subspace that is easier to handle, more structured, and more likely to yield promising solutions. Subspace search methods focus on identifying clusters within feature subsets, as clusters might not be apparent in the whole high-dimensional space due to the curse of dimensionality.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

Chen, M. S., Lin, J. Q., Wang, C. D., Huang, D., & Lai, J. H. (2025). Contrastive Ensemble Clustering. *IEEE Transactions on Neural Networks and Learning Systems*.

Farid, D. M., Nowe, A., & Manderick, B. (2016). An ensemble clustering for mining high-dimensional biological big data. *International Journal of Design & Nature and Ecodynamics*, *11*(3), 328-337.

Hong, X., Li, H., Miller, P., Zhou, J., Li, L., Crookes, D., ... & Zhou, H. (2019). Component-based feature saliency for clustering. *IEEE transactions on knowledge and data engineering*, *33*(3), 882-896.

Hu, G., & Rezaeipanah, A. (2025). Noise-robust semi-supervised clustering learning framework considering weighted consensus and pairwise similarities. *Neurocomputing*, *630*, 129700.

Li, J. (2010). *Ensemble clustering via heuristic optimisation* (Doctoral dissertation). Brunel University, School of Information Systems, Computing and Mathematics.

Li, S., Zhao, P., Wang, H., Wang, H., & Li, T. (2025). Neighbor self-embedding graph model for clustering ensemble. *Applied Soft Computing*, *171*, 112844.

Liu, Y., Li, S., & Tian, W. (2021, May). Autocluster: Meta-learning based ensemble method for automated unsupervised clustering. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 246-258). Cham: Springer International Publishing.

Mahmud, M. S., Zheng, H., Garcia-Gil, D., Garcia, S., & Huang, J. Z. (2025). RSPCA: Random Sample Partition and Clustering Approximation for ensemble learning of big data. *Pattern Recognition*, *161*, 111321.

Ren, S., Zhang, X., Li, H., Hu, C., & Chen, D. (2025). Advanced analysis of defect clusters in nuclear reactors using machine learning techniques. *Scientific Reports*, *15*(1), 22439.

Shang, Z., Dang, Y., Wang, H., & Liu, S. (2025). Representative Point-Based Clustering With Neighborhood Information for Complex Data Structures. *IEEE Transactions on Cybernetics*.

Tian, H. P., & Zhang, Z. (2025). Partial distance evidential clustering for missing data with multiple imputation. *Knowledge-Based Systems*, *310*, 112948.

Vukicevic, M., Radovanovic, S., Delibasic, B., & Suknovic, M. (2016). Extending meta-learning framework for clustering gene expression data with component-based algorithm design and internal evaluation measures. *International Journal of Data Mining and Bioinformatics*, *14*(2), 101-119.

Wei, Z., Wang, J., Zhao, Z., & Shi, K. (2025). Toward data efficient anomaly detection in heterogeneous edge–cloud environments using clustered federated learning. *Future Generation Computer Systems*, *164*, 107559.

Xu, P., Gao, H., & Wang, Y. (2025). Sparse dual-weighting ensemble clustering. *Cluster Computing*, *28*(2), 119.

Yang, C. H., Lee, B., Lee, Y. I., Chung, Y. F., & Lin, Y. D. (2025). An autoencoder-based arithmetic optimization clustering algorithm to enhance principal component analysis to study the relations between industrial market stock indices in real estate. *Expert Systems with Applications*, *266*, 126165.

Yu, Z., Zheng, X., Sun, J., Zhang, P., Zhong, Y., Lv, X., ... & Yang, J. (2025). Critical factors influencing live birth rates in fresh embryo transfer for IVF: insights from cluster ensemble algorithms. *Scientific Reports*, *15*(1), 3734.

Zhan, S., Jiang, H., & Shen, D. (2025). Co-regularized optimal high-order graph embedding for multi-view clustering. *Pattern Recognition*, *157*, 110892.

Zhang, Z., Chen, X., Wang, C., Wang, R., & Song, W. (2024). A Structured Bipartite Graph Learning Method for Ensemble Clustering. *A Structured Bipartite Graph Learning Method for Ensemble Clustering*.

Zheng, X., Lu, Y., Wang, R., Nie, F., & Li, X. (2025). Structured Graph-Based Ensemble Clustering. *IEEE Transactions on Knowledge and Data Engineering*.

Zhou, Z. F., Huang, D., & Wang, C. D. (2025). Pyramid contrastive learning for clustering. *Neural Networks*, *185*, 107217.