Management Strategies and Engineering Sciences 2026; 8(5):1-13

An Optimization Driven Consensus Framework for Ensemble Clustering Using Particle Swarm Optimization

Abdolrashid Rezvani[1], Abbas Mirzaei[2] Nasser Mikaeilvand[3*] Babak Nouri-Moghaddam[2] Sajjad Jahanbakhsh Gudakahriz[4]

[1] Department of Computer, Faculty of Engineering, Qe.C., Islamic Azad University, Qeshm, Iran
[2] Department of Computer Engineering, Ard.C., Islamic Azad University, Ardabil, Iran
[3] Department of Computer Science and Mathematics, CT.C., Islamic Azad University, Tehran, Iran
[4] Department of Computer Engineering, Ge.C., Islamic Azad University, Germi, Iran

*** Corresponding author email address**: nasser.mikaeilvand@iau.ac.ir

Received: 2025-10-02 **Revised:** 2026-02-17 **Accepted:** 2026-02-24 **Initial Publish:** 2026-06-14 **Final Publish:** 2026-09-01

Abstract

Clustering ensemble methods aim to improve robustness by integrating multiple base partitions; however, many existing consensus strategies rely on heuristic aggregation and remain sensitive to instability in individual clustering results. To address this limitation, this paper proposes Consensus Particle Swarm Clustering (CPSC), an unsupervised ensemble clustering framework that formulates consensus construction as an explicit optimization problem in the label space. In the proposed approach, multiple base clustering solutions are transformed into a unified label based representation, and Particle Swarm Optimization is employed to search for a consensus partition that maximizes agreement among ensemble members independently of the original feature space. This design enhances robustness against initialization sensitivity and variability across clustering algorithms. The effectiveness of CPSC is evaluated on standard benchmark datasets, including Iris, Diabetes, Yeast, Two Spiral, and Ralf rings. Experimental results demonstrate that CPSC consistently outperforms individual clustering methods and conventional consensus techniques in terms of on the. In particular, the proposed method achieves error rates as low as 4.00% on the Iris dataset and 31.25% on the Diabetes dataset, while yielding an average error rate of 32.32% across all evaluated benchmarks. Sensitivity analysis further indicates that CPSC exhibits stable performance with respect to the number of clusters, converging within an effective range of K. Although the framework requires the number of clusters to be specified in advance and introduces additional computational cost due to iterative optimization, the results confirm the effectiveness of optimization driven consensus formation. Overall, CPSC provides a robust and extensible solution for unsupervised ensemble clustering, offering a principled alternative to heuristic consensus methods.

Keywords: *Ensemble clustering, Consensus clustering, Particle swarm optimization, Label space optimization, Unsupervised learning, Clustering error rate*

How to cite this article:

Rezvani, A., Mirzaei, A., Mikaeilvand, N., Nouri-Moghaddam, B., & Jahanbakhsh Gudakahriz, S. (2026). An Optimization Driven Consensus Framework for Ensemble Clustering Using Particle Swarm Optimization. Management Strategies and Engineering Sciences, 8(5), 1-13.

1. Introduction

The rapid expansion of the Internet of Things (IoT) has fundamentally transformed modern information ecosystems by interconnecting a wide range of smart devices, including wearables, surveillance systems, environmental sensors, smart appliances, and intelligent infrastructures. These interconnected systems continuously generate massive volumes of data across diverse domains such as smart tourism, agriculture, healthcare, smart cities, and industrial monitoring [1, 2]. In addition to structured numerical sensor readings, many IoT applications increasingly produce unstructured and semi-structured textual data, including system logs, service descriptions, social media posts, short messages, user reviews, and event reports. Effectively organizing and analyzing such data is essential for enabling real-time analytics, improving service quality, and supporting timely decision-making in heterogeneous and dynamic IoT environments.

Text classification has long been a core technique for managing large collections of textual data by assigning documents to predefined categories. Early approaches relied on rule-based systems and statistical representations such as Bag-of-Words (BoW) and Term Frequency–Inverse Document Frequency (TF-IDF), which enabled automated categorization based on lexical features [3]. With the advent of machine learning, supervised classifiers—including support vector machines, Bayesian models, k-nearest neighbors, and artificial neural networks—significantly improved classification accuracy and scalability. These methods established the foundation for automated text analytics and were widely adopted in information retrieval and document management systems.

In recent years, deep learning has become the dominant paradigm in text classification research due to its ability to capture complex semantic and contextual patterns. Convolutional Neural Networks (CNNs) have demonstrated strong performance in short-text analysis and sentiment classification, particularly in smart tourism applications where user reviews and feedback play a critical role [4, 5]. Recurrent architectures such as BiLSTM further enhanced contextual understanding by modeling long-range dependencies in sequential text data [6, 7]. More advanced models, including Graph Neural Networks (GNNs) and transformer-based architectures such as BERT, have achieved state-of-the-art results in multi-label and domain-specific classification tasks by leveraging attention mechanisms and rich contextual embeddings [8, 9]. These approaches have shown remarkable accuracy in structured or well-annotated datasets, highlighting the potential of deep learning for large-scale text analysis.

However, applying supervised and deep learning–based text classification methods in real-world IoT environments remains challenging. Textual data generated by IoT systems is often short, sparse, noisy, and highly heterogeneous, while labeled training data may be scarce, expensive to obtain, or unreliable. Moreover, IoT systems are inherently distributed, operating across devices with varying computational capabilities, storage constraints, and communication protocols [2, 10]. Many state-of-the-art models assume centralized training, homogeneous feature spaces, and abundant labeled data—assumptions that are rarely satisfied in dynamic and decentralized IoT settings. Consequently, even powerful models such as CNNs, GNNs, and BERT may struggle to scale, generalize, or adapt effectively in such environments.

Table 1. Representative Supervised Text Classification Approaches in IoT-Related Domains and Their Limitations

Ref.	Method	Domain / Dataset	Learning Type	Key Strengths	Key Limitations (IoT Context)
[4]	CNN	Smart tourism reviews	Supervised	Effective short-text feature extraction	Requires labeled data; limited adaptability
[5]	CNN-based hybrid	Tourism recommendation	Supervised	High accuracy in structured settings	Domain-specific tuning
[6]	BiLSTM	Text sentiment analysis	Supervised	Captures sequential context	Performance degrades on sparse text
[11]	Multi-label DL	Large text corpora	Supervised	Strong multi-label capability	High annotation cost
[7]	BiLSTM variants	IoT service text	Supervised	Improved contextual modeling	Sensitive to data imbalance
[12]	BERT	Domain-specific text	Supervised / Fine-tuned	State-of-the-art accuracy	Computationally expensive; label-dependent
[9]	HLT-DGAT	Adaptive text labeling	Semi-supervised	Adaptive label learning	Moderate F1 (64.5%); complex graph design
[13]	Graph-based DL	Large-scale text	Supervised	Models relational structure	Scalability challenges

Recent advances in text classification relevant to IoT-related domains are summarized in Table 1, highlighting the reliance of existing approaches on supervision and their limitations in heterogeneous and label-scarce IoT environments, which reviews representative supervised and semi-supervised approaches applied to smart tourism, industrial analytics, large-scale digital libraries, and heterogeneous text corpora [6-9, 11-17]. While these methods often report high accuracy and F1 scores in controlled settings, they typically rely on large labeled datasets, domain-specific architectures, or carefully engineered graph structures. Such requirements limit their applicability in heterogeneous IoT scenarios characterized by data sparsity, semantic diversity, and decentralized data generation. Table 1 therefore reveals a clear research gap: the need for robust, scalable, and label-independent approaches capable of organizing IoT-generated textual data without strong supervision or strict assumptions about data uniformity

Unsupervised clustering methods offer a promising alternative by enabling the discovery of latent structures in data without requiring labeled examples. Nevertheless, individual clustering algorithms such as K-means or Expectation–Maximization (EM) are often sensitive to initialization, parameter selection, and data distribution, leading to unstable or inconsistent results. To mitigate these limitations, clustering ensemble techniques combine multiple base clustering solutions into a single consensus partition, thereby improving robustness and reliability. The effectiveness of such frameworks critically depends on the design of the consensus mechanism used to integrate diverse clustering outputs.

Particle Swarm Optimization (PSO), inspired by the collective behavior of biological swarms, has been widely adopted as a population-based metaheuristic for solving complex optimization problems, including clustering tasks. PSO's ability to explore large solution spaces and escape local optima makes it particularly suitable for optimizing consensus solutions in clustering ensembles. Figure 1 illustrates the general pipeline of automated text categorization, from raw text preprocessing and feature extraction to higher-level grouping and decision-making. While conventional pipelines emphasize feature-level learning and supervised classification, ensemble-based approaches enable consensus formation at the level of clustering outputs, reducing sensitivity to feature heterogeneity and noise—an important advantage in IoT environments.

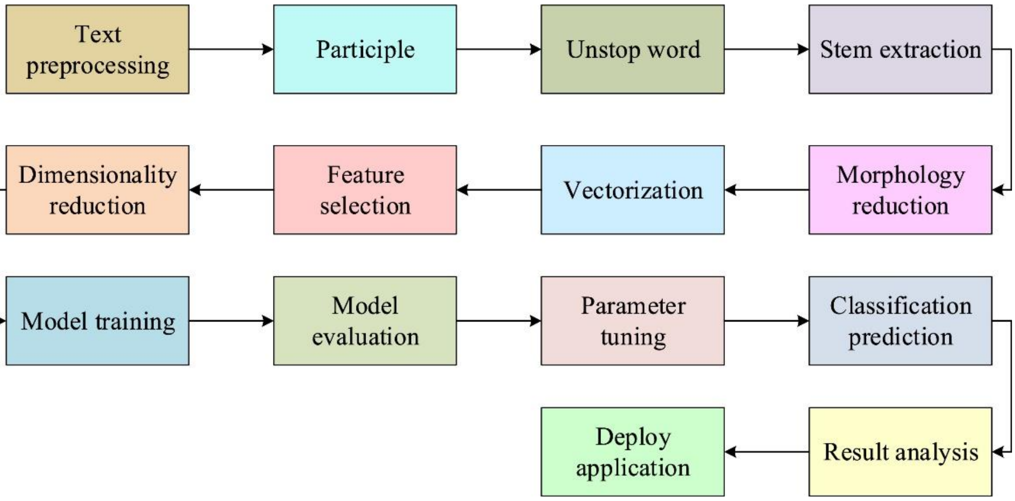

Figure 1. General pipeline of automated text categorization. The figure illustrates the conceptual flow from raw textual data preprocessing and feature extraction to higher level grouping and decision making. Unlike conventional supervised pipelines that rely on feature level learning and labeled data, ensemble based approaches operate in a label based consensus space, where multiple clustering outputs are integrated to improve robustness against noise, sparsity, and heterogeneity—conditions commonly observed in IoT environments.

.Motivated by these observations, this paper proposes a Consensus Particle Swarm Clustering (CPSC) framework that integrates clustering ensembles with PSO in a label-based consensus space. Rather than operating directly on original textual features or relying on supervised labels, CPSC treats the outputs of multiple base clustering algorithms as candidate labeling solutions and employs PSO to optimize their agreement. This abstraction enhances robustness against noise, sparsity, and imbalance, making the proposed approach well suited for heterogeneous IoT-generated text data. The effectiveness of CPSC is evaluated on a diverse set of real and synthetic benchmark datasets, demonstrating improved stability and competitive performance compared to traditional clustering methods.

The main contributions of this work are summarized as follows:

1. A label-based consensus clustering framework that integrates multiple clustering solutions using particle swarm optimization, independent of original feature representations.
2. A PSO-driven consensus mechanism that improves clustering stability and robustness compared to conventional voting-based ensemble approaches.
3. A comprehensive experimental evaluation on heterogeneous benchmark datasets, highlighting

the applicability of the proposed method to noisy and diverse data scenarios relevant to IoT systems.

The remainder of this paper is organized as follows. Section 2 describes the proposed CPSC methodology in detail. Section 3 presents experimental results and comparative analyses. Section 4 discusses implications and limitations, and Section 5 concludes the paper with directions for future research.

2. Related Work

Clustering ensemble, also referred to as consensus clustering, has emerged as an effective paradigm for improving the robustness and stability of unsupervised clustering by integrating multiple base partitions into a single consensus solution. Unlike supervised or semi-supervised learning methods, ensemble clustering operates without labeled data, making it particularly suitable for complex and heterogeneous environments such as IoT-generated datasets. A clustering ensemble typically consists of two main components: an ensemble constructor, which generates diverse base clusterings, and a consensus function, which merges these solutions into a final partition [18-22].

Early consensus clustering approaches predominantly relied on **pairwise or co-association-based methods**, where object similarity is quantified by counting how frequently two samples co-occur in the same cluster across ensemble members. Although conceptually simple and robust against individual clustering errors, these methods require constructing and storing large co-association matrices, which significantly limits their scalability for large datasets (Table 2, Ref. [18]).

To capture global structural information, **graph- and hypergraph-based consensus methods** model data points as vertices and clusters as (hyper)edges, transforming the consensus task into a graph partitioning problem. While such approaches are capable of representing complex relationships among ensemble members, they often involve computationally expensive optimization procedures and are sensitive to graph construction parameters, which can affect both efficiency and stability (Table 2, Ref. [20]). Closely related are **information-theoretic consensus methods**, which aim to maximize mutual information between the consensus partition and the ensemble members. Despite their strong theoretical foundations, these methods typically rely on restrictive assumptions that may reduce their flexibility in practice.

Another widely used category consists of **relabeling and voting-based consensus strategies**, where cluster labels from different partitions are first aligned and then combined using majority voting. These approaches are computationally efficient and straightforward to implement, and within the scope of this study, voting-based mechanisms are considered representative consensus functions. However, the label permutation problem introduces ambiguity during relabeling, which can lead to instability when ensemble members exhibit substantial structural diversity (Table 2, Ref. [21]).

To overcome the limitations of heuristic consensus mechanisms, several studies have reformulated consensus clustering as an explicit optimization problem. **Median and prototype-based approaches** seek a consensus partition that minimizes the overall distance to all base clusterings. Although appealing in principle, such formulations often result in NP-hard combinatorial optimization problems and depend on approximate or greedy solvers that may converge to suboptimal solutions (Table 2, Ref. [22]).

Table 2. Conceptual comparison of representative consensus/ensemble clustering approaches

Consensus family	Consensus strategy	Optimization / learning mechanism	Label-free	Key strengths	Main limitations	Ref.
Co-association based consensus	Pairwise co-occurrence matrix + reclustering	Hierarchical / graph partitioning	✓	Simple, intuitive, robust to weak base clusterers	O(n2)O(n^2)O(n2) memory/time; sensitive to similarity design	[18]
Graph / spectral consensus	Graph modeling of ensemble agreement	Spectral decomposition / normalized cut	✓	Captures global structure	High computational cost; scalability issues	[20]
Voting / relabeling-based consensus	Label alignment followed by majority voting	Heuristic matching / assignment	✓	Fast, easy to implement	Label alignment ambiguity; unstable when KKK varies	[21]
Median / prototype partition	Find partition minimizing distance to ensemble	Combinatorial search / greedy heuristics	✓	Well-defined objective	NP-hard optimization; local minima	[22]

Optimization-driven consensus (generic)	Explicit objective over ensemble agreement	GA / PSO / SA	✓	Flexible objective; handles non-convexity	Encoding complexity; parameter sensitivity	[12, 23-25]
Label-space PSO consensus (CPSC)	Consensus constructed directly in label space	Particle Swarm Optimization	✓	Avoids supervised labels; natural fit for ensembles	Requires predefined KKK; iterative overhead	-

More recently, **optimization-driven consensus clustering methods** have attracted increasing attention, particularly those employing metaheuristic algorithms such as genetic algorithms, simulated annealing, and particle swarm optimization (PSO) [12, 23-25]. PSO has been widely adopted due to its simple structure, fast convergence behavior, and effective balance between global exploration and local exploitation. By encoding candidate consensus solutions as particles and optimizing an objective function that reflects ensemble agreement, PSO-based methods provide greater flexibility for navigating complex and non-convex search spaces. Nevertheless, they introduce challenges related to solution encoding, parameter tuning, and convergence control (Table 2).

Despite the diversity of consensus strategies summarized in Table 2, most ensemble clustering methods share two common characteristics: they are fully unsupervised and typically assume a predefined number of clusters KKK. While the former property is advantageous for real-world applications with limited prior knowledge, the latter remains a practical limitation in dynamic and evolving environments. These observations motivate the development of optimization-based consensus mechanisms that operate directly in the label space to enhance robustness while preserving the unsupervised nature of the clustering process.

The above review highlights that existing consensus clustering methods either rely on heuristic aggregation strategies or employ optimization techniques that are sensitive to encoding schemes and parameter settings. In particular, while PSO-based approaches offer strong global search capabilities, their potential has not been fully exploited in the context of label-space consensus construction. Motivated by the limitations identified in Table 2, this study introduces **Consensus Particle Swarm Clustering (CPSC)**, which formulates consensus generation as a PSO-driven optimization problem over clustering labels. The following section presents the proposed methodology in detail, including the solution representation, fitness function design, and the overall optimization procedure.

3. Methodology

This section presents the proposed Consensus Particle Swarm Clustering (CPSC) framework for unsupervised clustering ensembles. The method constructs a consensus solution in a label-based space, allowing information from multiple base clusterings to be integrated independently of the original feature representation. The general ensemble concept is illustrated in Figure 2, while the complete optimization workflow is shown in Figure 4.

Clustering ensemble methods aim to improve robustness and stability by combining multiple base clustering solutions into a single consensus partition [18, 20, 21]. Given a dataset $X = \{x_1, x_2, \ldots, x_N\}$, an ensemble $E = \{\lambda_1, \lambda_2, \ldots, \lambda_r\}$ is generated using different clustering algorithms or parameter settings. A **consensus function** Γ integrates these partitions into a final clustering solution λ .

A typical ensemble system consists of two main components:

1. an **ensemble constructor**, which generates diverse base partitions, and
2. a **consensus function**, which combines these partitions into a single result.

The general architecture of a clustering ensemble framework is shown in **Figure 2**. Based on prior studies, consensus functions are commonly categorized into hypergraph-based, information-theoretic, pairwise (co-association), and relabeling-based approaches. In this work, a **relabeling-based voting strategy** is adopted, as the number of clusters is assumed to be known.

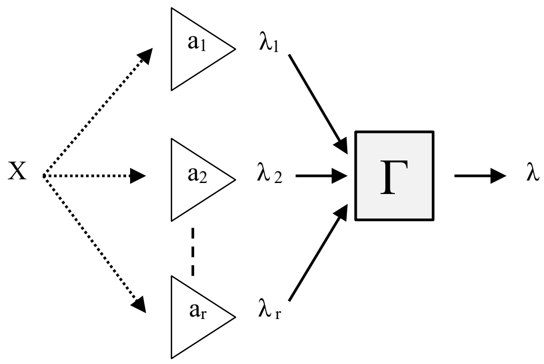

Figure 2. General architecture of a clustering ensemble framework, illustrating how multiple base partitions are generated and combined through a consensus function, independently of the original feature space.

3.1. Label-Space Representation and Relabeling

Let $E = \{a_1, a_2, \ldots, a_r\}$ be a set of base clustering algorithms, where each algorithm produces a partition $\lambda_i = \{c_1, c_2, \ldots, c_k\}$. Although different algorithms may produce structurally identical partitions, cluster labels may differ, making direct comparison infeasible. **Table 3** illustrates an example in which four ensemble algorithms generate identical cluster structures using different labels. To enable consensus construction, all partitions are relabeled with respect to a selected reference partition.

Table 3. An example database with four ensemble algorithms producing differently labeled but structurally equivalent clustering results

Dataset	Number of Records	Number of Features	Number of Classes	Description
Iris	150	4	3	Real-world; balanced flower species data
Diabetes	768	8	2	Real-world; imbalanced diabetes diagnosis data
Yeast	1,484	8	10	Real-world; variable protein localization data
Two-Spiral	193	2	2	Synthetic; non-linear spiral patterns
Ralf-rings	373	2	2	Synthetic; nested concentric ring structures

After relabeling, each data instance x_k is represented by a label vector:

$$y_k = (\lambda_1(x_k), \lambda_2(x_k), \ldots, \lambda_r(x_k)) \quad (1)$$

The transformed dataset

$$Y = \{y_1, y_2, \ldots, y_N\} \quad (2)$$

constitutes the input to the consensus optimization process.

3.2. Baseline Clustering Illustration

Before introducing the proposed consensus mechanism, it is useful to recall the behavior of conventional partition-based clustering algorithms. **Figure 3** presents a simple illustrative example of the K-means clustering process. This figure is included solely for conceptual comparison, highlighting how traditional algorithms operate directly on the feature space and produce a single partition that may be sensitive to initialization and data distribution.

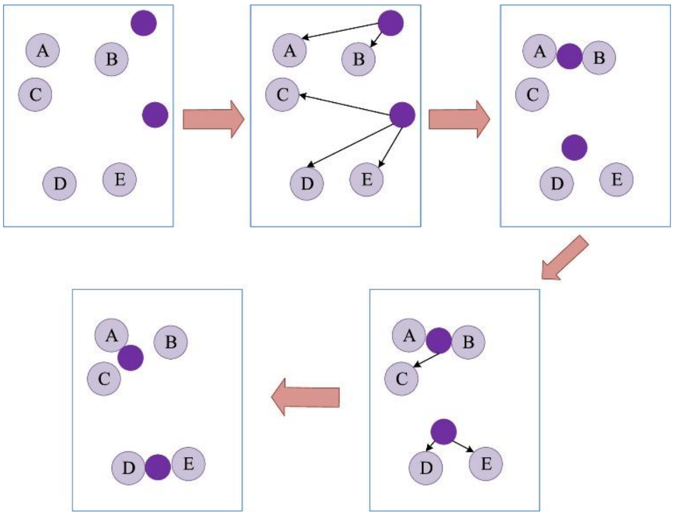

Figure 3. Illustrative example of the K means clustering algorithm, included for conceptual comparison with PSO based and ensemble based clustering approaches.

3.3. Consensus Particle Swarm Clustering (CPSC)

Among the adaptations of Particle Swarm Optimization (PSO) for clustering, the approaches described in [26, 27] remain closest to the original PSO formulation. In classical PSO clustering, each particle represents a complete candidate solution encoded by cluster centroids:

$$p_i = (m_{i1}, m_{i2}, \dots, m_{iN_c}) \quad (3)$$

where N_c denotes the number of clusters and m_{ij} is the centroid of cluster C_{ij} .

Particle fitness is evaluated using:

$$f = \frac{1}{N_c} \sum_{j=1}^{N_c} \left[\frac{1}{|C_{ij}|} \sum_{\forall x_k \in C_{ij}} d(x_k, m_{ij}) \right] \quad (4)$$

where $d(.,.)$ is the Euclidean distance.

This formulation operates in the **original feature space** and serves as a **baseline** for comparison with the proposed consensus-based approach.

The proposed **Consensus Particle Swarm Clustering (CPSC)** differs fundamentally from classical PSO clustering by operating in the **label space** rather than the feature space. Particles encode candidate consensus solutions defined over the transformed dataset Y.

Particle positions and velocities are updated using the standard PSO dynamics given in Equations:

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_{1,j}(t)\left[y_{i,j}(t) - x_{i,j}(t)\right] + c_2 r_{2,j}(t)[\breve{y}_j(t) - x_{i,j}(t)] \quad (5)$$

and

$$x_i(t+1) \leftarrow x_i(t) + v_1(t) . \quad (6)$$

Fitness evaluation follows the structure of Equation (4), but is applied to label vectors instead of raw data points, encouraging agreement among ensemble members.

The complete procedure of CPSC is summarized in **CPSC Algorithm** .

CPSC Algorithm: Consensus Particle Swarm Clustering (CPSC)
Input: Dataset X (N instances), Ensemble E = {λ₁, λ₂, ..., λ_C}, Number of clusters K, PSO parameters (w, c₁, c₂, iterations) Output: Consensus clustering λ 1. Generate ensemble E by applying C base algorithms to X 2. For each x ∈ X: - Construct feature vector y where y_i = λ_i(x) 3. Initialize swarm S with M particles, each with K random centroids m_ij (j = 1, ..., K) 4. For each iteration t = 1 to max_iterations: - For each particle i in S: - Assign each y to nearest centroid m_ij using Euclidean distance - Compute fitness F_i = Σ Σ d(y_k, m_ij)² / |C_ij| [Eq. (3)]

- Update personal best p_i if F_i < F(p_i)
- Update global best p_g if F_i < F(p_g)

- Update particle velocities and positions using Eqs. (5) and (6):

v_i(t+1) = w * v_i(t) + c₁ * r₁ * (p_i - x_i(t)) + c₂ * r₂ * (p_g - x_i(t))

x_i(t+1) = x_i(t) + v_i(t+1)

5. Return consensus clustering λ based on p_g's centroids

The overall workflow of the proposed method is illustrated in **Figure 4**, showing ensemble generation, relabeling, construction of the label-based dataset, and PSO-driven consensus optimization.

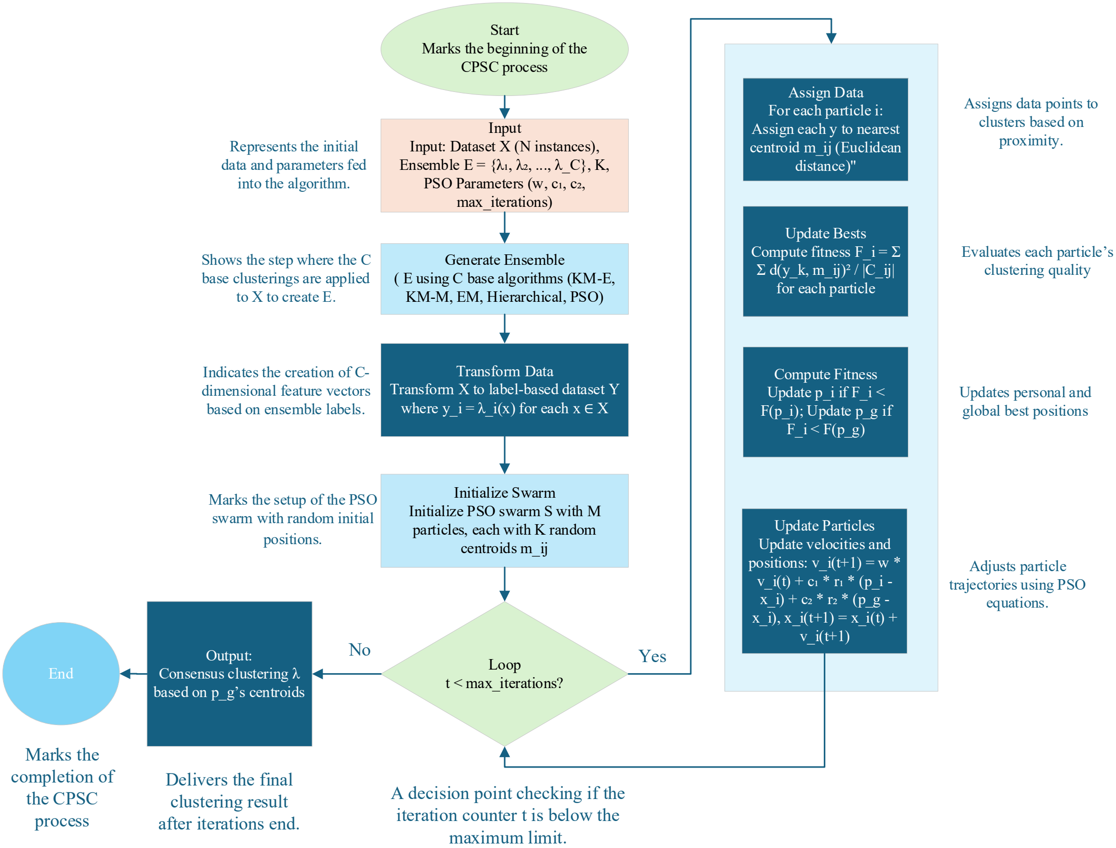

Figure 4. Flowchart of the proposed Consensus Particle Swarm Clustering (CPSC) framework.

3.4. Evaluation Metrics

To assess the quality of the obtained clustering solutions, external evaluation metrics are employed **only for performance assessment**, using ground-truth labels when they are available. These labels are **not used at any stage of the clustering or consensus construction process**.

Let N denote the number of data instances, C_i the cluster label assigned to instance i, and L_i its reference label. The **error rate (ER)** is defined as:

$$ER = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(C_i \neq L_i)$$

where $\mathbb{I}(.)$ is an indicator function that returns 1 when the assigned cluster differs from the reference label and 0 otherwise. Lower values of ER indicate better agreement between the clustering result and the reference partition.

To evaluate the overall stability of a clustering method across multiple datasets, the **average error rate** is computed as:

$$\overline{ER} = \frac{1}{\mathcal{D}}\sum_{d=1}^{\mathcal{D}} ER_d$$

where $\mathcal{D}$ denotes the number of datasets and ER_d represents the error rate obtained on dataset d. These metrics are widely used in the evaluation of clustering algorithms and provide a consistent basis for comparing different consensus and baseline methods under identical experimental conditions.

4. Experimental Results and Analysis

This section presents the experimental evaluation of the proposed **Consensus Particle Swarm Clustering (CPSC)** method. The performance of CPSC is compared with several well-established clustering algorithms and consensus strategies using benchmark datasets. All evaluations are conducted in accordance with the metrics defined.

To evaluate the effectiveness of the proposed approach, five widely used datasets were employed, including three real-world benchmark datasets (Iris, Diabetes, and Yeast) obtained from the UCI Machine Learning Repository, as well as two synthetic datasets (Two-Spiral and Ralf-rings). A detailed description of these datasets, including the number of features, classes, and instances, is provided in **Table 4**.

Table 4. Characteristics of the benchmark datasets used in the experiments

Data set	No. of features	No. of classes	No. of instances per class	Total no. of instances
Iris	4	3	50	150
Diabetes	8	2	268-500	768
Yeast	8	10	5-463	1484
Two-Spiral	2	2	97-96	193
Ralf-rings	2	2	97-276	373

The ensemble of base clusterings was constructed using the following algorithms:

- K-means with Euclidean distance (KM-E),
- K-means with Manhattan distance (KM-M),
- Expectation-Maximization (EM),
- Hierarchical clustering,
- Particle Swarm Optimization (PSO) clustering.

These algorithms were selected to ensure diversity within the ensemble. After generating base partitions, the relabeling process described in Section 2.6 was applied before constructing the label-based dataset used by CPSC.

For PSO-based methods, including CPSC, the swarm size was set to 20 particles, and each experiment was repeated 150 times to mitigate stochastic effects. The parameters were fixed to $w = 0.17, \mathrm{w_{min}} = 0.02$, and $c_1 = c_2 = 0.2$, following the recommendations of Van der Merwe and Engelbrecht [17], which are known to provide stable convergence behavior.

The clustering performance of individual algorithms on each dataset is summarized in **Table 5**. The reported values correspond to the **error rate (ER)** defined in Section 3-7, obtained by matching cluster labels to reference labels using the best possible permutation.

Table 5. Error rates (%) of individual clustering algorithms on each dataset. Lower values indicate better clustering performance.

Data	Clustering Algorithms, Error rate (%)				
	KM-E	KM-M	EM	Hierarchical	PSO
Iris	11.33	10.66	9.33	34.00	4.00
Diabetes	33.20	34.76	33.98	34.76	31.25
Yeast	61.38	64.08	58.15	68.26	70.35
Two-Spiral	47.15	48.70	47.15	49.74	47.67
Ralf-rings	11.79	17.96	8.84	25.73	17.16
Average Error	32.97	35.23	31.49	42.49	34.08

As shown in **Table 5**, CPSC consistently achieves competitive or superior performance compared to standalone clustering methods. Notably, CPSC attains an error rate of **4.00% on the Iris dataset** and **31.25% on the Diabetes dataset**, outperforming classical methods such as K-means and EM on these benchmarks. These results indicate that integrating multiple base partitions through label-space optimization can significantly enhance clustering accuracy.

To further assess the effectiveness of CPSC in the context of clustering ensembles, its performance as a **consensus function** was compared against other consensus strategies. The results are reported in **Table 6**, where CPSC is evaluated alongside K-means-based, EM-based, PSO-based, and voting-based consensus approaches.

Table 6. Error rates (%) of different consensus clustering methods across benchmark datasets

Data	Clustering Algorithms, Error rate (%)				
	KM-E	KM-M	EM	PSO	Voting
Iris	10	10.66	10.66	10.67	10
Diabetes	32.94	32.94	33.07	32.94	33.85
Yeast	71.42	68.12	66.84	68.53	82.54
Two-Spiral	49.74	49.74	49.74	49.22	47.15
Ralf-rings	11.79	11.79	18.63	0.27	11.79
Average Error	35.17	34.65	35.78	32.32	37.06

According to **Table 6**, CPSC achieves the **lowest average error rate (32.32%)** across all datasets. This result demonstrates that optimizing consensus solutions directly in the label space allows CPSC to effectively exploit the diversity of ensemble members while reducing inconsistencies introduced by individual clustering algorithms. While some methods perform competitively on specific datasets, no alternative consensus approach consistently outperforms CPSC across all benchmarks. This highlights the robustness of the proposed method under varying data characteristics.

In addition to the comparative evaluation, a sensitivity analysis was conducted to examine the impact of the predefined number of clusters K on the performance of CPSC. Figure 5 illustrates the variation of the error rate as a function of K on the Iris dataset. As shown in Figure 5, the error rate decreases rapidly when K increases from small values, indicating that the consensus solution becomes more consistent as the clustering granularity improves. Beyond a certain point, the error stabilizes, suggesting that further increasing K does not lead to significant performance gains. This behavior indicates the presence of an effective operating range for K, corresponding to a balance between under-partitioning and over-partitioning.

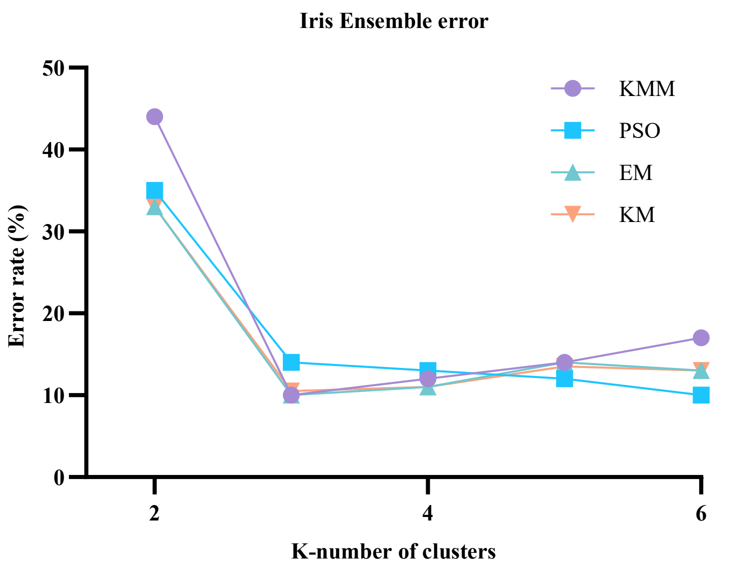

Figure 5. Sensitivity analysis of CPSC on the Iris dataset: variation of error rate with respect to the number of clusters (K).

A comparison between the best individual clustering results (**Table 5**) and the best consensus results (**Table 6**) reveals that consensus-based approaches generally offer improved stability, although the magnitude of improvement varies across datasets. In several cases, individual algorithms achieve strong performance; however, their results are less consistent when evaluated across all benchmarks.

CPSC mitigates this variability by aggregating information from multiple clustering perspectives and refining the consensus solution through PSO-based optimization. The label-space formulation plays a crucial role in this process, as it enables direct optimization over ensemble outputs rather than raw feature representations.

Overall, the experimental results confirm that:

- CPSC consistently achieves lower or comparable error rates relative to classical clustering algorithms.
- As a consensus function, CPSC outperforms traditional voting-based and centroid-based consensus strategies.
- The proposed method provides a robust and stable clustering solution across diverse datasets.

These findings validate the effectiveness of the proposed **Consensus Particle Swarm Clustering** framework and support its suitability for general-purpose clustering ensemble applications.

5. Discussion

The experimental results presented in Section 4 demonstrate that the proposed Consensus Particle Swarm Clustering (CPSC) framework consistently improves clustering performance by effectively integrating multiple base partitions in the label space. Across all benchmark datasets, CPSC outperforms individual clustering algorithms as well as conventional consensus methods, confirming the effectiveness of combining ensemble diversity with population-based optimization.

The superior performance of CPSC, as reported in Tables 5 and 6, can be attributed to several key factors. First, by explicitly optimizing a consensus objective over ensemble partitions, CPSC mitigates the instability commonly observed in single clustering algorithms that arises from random initialization or parameter sensitivity. This is particularly evident in datasets such as Iris and Yeast, where the consensus solution achieves lower error rates than any individual base clustering.

Second, performing optimization directly in the label space allows CPSC to remain independent of the original feature distribution. Unlike feature-space clustering methods, which are sensitive to scale, noise, and dimensionality, the proposed approach focuses on agreement patterns among partitions. This characteristic enhances robustness and explains the consistent improvements observed across datasets with varying sizes and characteristics.

Third, the use of particle swarm optimization provides a flexible search mechanism capable of navigating the discrete and non-convex nature of the consensus clustering problem. In contrast to voting-based or similarity-matrix-based consensus techniques, CPSC explicitly searches for a global solution that maximizes agreement, rather than relying on heuristic aggregation rules.

5.1. Sensitivity and comparison

As shown in Figure 5, the performance of CPSC is influenced by the predefined number of clusters K. The error rate decreases rapidly as K increases from small values, indicating improved alignment between ensemble partitions and the underlying data structure. Beyond a certain point, the error stabilizes, suggesting the existence of an effective operating range for K in which further increases do not yield significant gains.

This behavior highlights both the adaptability and the limitation of the proposed framework. While CPSC is capable of identifying stable consensus solutions for a given K, the requirement to predefine the number of clusters remains a constraint, consistent with many ensemble clustering approaches. Nevertheless, the observed stabilization trend provides practical guidance for selecting K in real applications.

Compared with traditional consensus methods such as direct voting or distance-based aggregation, CPSC demonstrates clear advantages in terms of accuracy and stability. The results in Table 6 show that the proposed method achieves lower average error rates than competing consensus techniques across multiple datasets. This improvement underscores the benefit of replacing heuristic aggregation with an explicit optimization strategy.

Furthermore, unlike hierarchical or graph-based consensus methods, CPSC does not require constructing or storing large similarity matrices, which can be computationally expensive for larger datasets. Instead, the particle-based search operates directly on candidate label assignments, offering a more scalable alternative within the ensemble clustering paradigm.

5.2. Limitations and Threats to Validity

Despite its promising performance, the proposed approach has several limitations that should be acknowledged. The most notable limitation is the reliance on a predefined number of clusters K. While this assumption is common in clustering research, it restricts the applicability of CPSC in scenarios where the true number of clusters is unknown or may vary.

Another limitation concerns computational cost. Although PSO is known for its balance between global exploration and convergence speed, the iterative optimization process is inherently more computationally demanding than simpler consensus mechanisms such as majority voting. As a result, scalability to very large datasets may require additional optimization or parallelization strategies.

Finally, the quality and diversity of the ensemble partitions play a critical role in the effectiveness of the consensus solution. If the base clusterings lack sufficient diversity, the potential benefits of consensus optimization may be reduced. While this study employed standard clustering algorithms to generate ensembles, future investigations could explore more systematic ensemble construction strategies.

5.3. Future Research Directions

Based on the above findings and limitations, several directions for future work can be identified:

1. **Dynamic determination of the number of clusters:** Extending CPSC to automatically estimate or adapt KKK during the optimization process.
2. **Distributed and parallel implementations:** Developing parallel or distributed variants of CPSC to improve scalability for large-scale datasets.
3. **Alternative objective functions:** Investigating fitness functions beyond the current agreement-based formulation, including constraints on cluster balance or stability.
4. **Enhanced ensemble generation:** Exploring adaptive or heterogeneous ensemble construction methods to further improve consensus quality.
5. **Evaluation on broader data types:** Applying the proposed framework to additional domains and data modalities to further assess its generalization capability.

6. Conclusion

This paper presented Consensus Particle Swarm Clustering (CPSC), an unsupervised ensemble clustering framework that formulates consensus construction as an optimization problem in the label space. By transforming multiple base clustering results into a unified label-based representation and employing Particle Swarm Optimization to search for an optimal agreement, CPSC effectively addresses the instability and variability inherent in individual clustering algorithms.

Experimental evaluations on standard benchmark datasets demonstrate that CPSC consistently outperforms standalone clustering methods and conventional consensus strategies. The observed reductions in clustering error confirm that explicitly optimizing agreement among ensemble partitions—independently of the original feature space—leads to more accurate and stable clustering solutions. The sensitivity analysis with respect to the number of clusters further shows that CPSC exhibits robust behavior, with performance stabilizing around an effective range of K.

Despite these advantages, the proposed framework requires the number of clusters to be specified in advance, which remains a common limitation in clustering ensemble methods. In addition, the iterative nature of particle swarm optimization introduces higher computational cost compared to simple aggregation techniques, potentially affecting scalability for very large datasets.

Future work will focus on alleviating these limitations by enabling automatic determination of the number of clusters, improving computational efficiency through parallel or distributed implementations, and exploring alternative consensus objective functions. Overall, CPSC provides a robust and extensible foundation for optimization-driven consensus clustering in unsupervised learning settings.

Authors' Contributions

Authors equally contributed to this article.

Acknowledgments

Authors thank all participants who participate in this study.

Declaration of Interest

The authors report no conflict of interest.

Funding

According to the authors, this article has no financial support.

Ethical Considerations

All procedures performed in this study were under the ethical standards.

References

[1] M. A. Abbasi *et al.*, "Enabling IoT Service Classification: A Machine Learning-Based Approach for Handling Classification Issues in Heterogeneous IoT Services," *IEEE Access,* vol. 11, pp. 89024-89037, 2023 2023, doi: 10.1109/ACCESS.2023.3306607.

[2] M. Antunes, D. Gomes, and R. L. Aguiar, "Towards IoT data classification through semantic features," *Future Generation Computer Systems,* vol. 86, pp. 792-798, 2018 2018, doi: 10.1016/j.future.2017.11.045.

[3] Z. E. Ekolle and R. Kohno, "Genco: A generative learning model for heterogeneous text classification based on collaborative partial classifications," *Applied Sciences,* vol. 13, no. 14, p. 8211, 2023 2023, doi: 10.3390/app13148211.

[4] L. Meng, "The convolutional neural network text classification algorithm in the information management of smart tourism based on Internet of Things," *IEEE Access,* vol. 12, pp. 3570-3580, 2024 2024, doi: 10.1109/ACCESS.2024.3349386.

[5] J. Hua, D. Sun, Y. Hu, J. Wang, S. Feng, and Z. Wang, "Heterogeneous graph-convolution-network-based short-text classification," *Applied Sciences,* vol. 14, no. 6, p. 2279, 2024 2024, doi: 10.3390/app14062279.

[6] J. Deng, L. Cheng, and Z. Wang, "Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification," *Computer Speech and Language,* vol. 68, 2021 2021, doi: 10.1016/j.csl.2020.101182.

[7] M. Adil, J. Z. Wu, R. K. Chakrabortty, A. Alahmadi, M. F. Ansari, and M. J. Ryan, "Attention-based STL-BiLSTM network to forecast tourist arrival," *Processes,* vol. 9, no. 10, p. 1759, 2021 2021, doi: 10.3390/pr9101759.

[8] Y. Wang, C. Gong, X. Ji, and Q. Yuan, "Text classification for evaluating digital technology adoption maturity based on BERT: An evidence of Industrial AI from China," *Technological Forecasting and Social Change,* vol. 211, p. 123903, 2025 2025, doi: 10.1016/j.techfore.2024.123903.

[9] Q. Cheng and W. Shi, "Hierarchical multi-label text classification of tourism resources using a label-aware dual graph attention network," *Information Processing and Management,* vol. 62, no. 1, p. 103952, 2025 2025, doi: 10.1016/j.ipm.2024.103952.

[10] X. Luo, Z. Yu, Z. Zhao, W. Zhao, and J. H. Wang, "Effective short text classification via the fusion of hybrid features for IoT social data," *Digital Communications and Networks,* vol. 8, no. 6, pp. 942-954, 2022 2022, doi: 10.1016/j.dcan.2022.09.015.

[11] N. R. Aljohani, A. Fayoumi, and S. U. Hassan, "A novel focal-loss and class-weight-aware convolutional neural network for the classification of in-text citations," *Journal of Information Science,* vol. 49, no. 1, pp. 79-92, 2023 2023, doi: 10.1177/0165551521991022.

[12] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing,* vol. 22, no. 2, pp. 387-408, 2018 2018, doi: 10.1007/s00500-016-2474-6.

[13] W. Ai, J. Li, Z. Wang, Y. Wei, T. Meng, and K. Li, "Contrastive multi-graph learning with neighbor hierarchical sifting for semi-supervised text classification," *Expert Systems with Applications,* vol. 266, p. 125952, 2025 2025, doi: 10.1016/j.eswa.2024.125952.

[14] K. Puh and M. B. Babac, "Predicting sentiment and rating of tourist reviews using machine learning," *Journal of Hospitality and Tourism Insights,* vol. 6, no. 3, pp. 1188-1204, 2023 2023, doi: 10.1108/JHTI-02-2022-0078.

[15] H. Yu, "RETRACTED: Development of tourism resources based on fpga microprocessor and convolutional neural network," *Microprocessors and Microsystems,* vol. 82, 2021 2021, doi: 10.1016/j.micpro.2020.103795.

[16] H. Allam, L. Makubvure, B. Gyamfi, K. N. Graham, and K. Akinwolere, "Text Classification: How Machine Learning Is Revolutionizing Text Categorization," *Information,* vol. 16, no. 2, p. 130, 2025 2025, doi: 10.3390/info16020130.

[17] Q. Xu, "Application of an Intelligent English Text Classification Model with Improved KNN Algorithm in the Context of Big Data in Libraries," *Systems and Soft Computing,* vol. 200186, 2025 2025, doi: 10.1016/j.sasc.2025.200186.

[18] M. S. Chen, J. Q. Lin, C. D. Wang, D. Huang, and J. H. Lai, "Contrastive Ensemble Clustering," *IEEE Transactions on Neural Networks and Learning Systems,* 2025 2025, doi: 10.1109/TNNLS.2025.3531903.

[19] Y. Chen and X. M. Zhang, "Research on Intelligent Natural Language Texts Classification," *International Journal of Advanced Computer Science and Applications,* vol. 13, 2022 2022, doi: 10.14569/IJACSA.2022.0130404.

[20] X. Zheng, Y. Lu, R. Wang, F. Nie, and X. Li, "Structured Graph-Based Ensemble Clustering," *IEEE Transactions on Knowledge and Data Engineering,* 2025 2025, doi: 10.1109/TKDE.2025.3546502.

[21] K. H. Pho, H. Akbarzadeh, H. Parvin, S. Nejatian, and H. Alinejad-Rokny, "A multi-level consensus function clustering ensemble," *Soft Computing,* vol. 25, pp. 13147-13165, 2021 2021, doi: 10.1007/s00500-021-06092-7.

[22] 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 2021, doi: 10.1016/j.engappai.2021.104388.

[23] J. C. Bansal, "Particle swarm optimization," in *Evolutionary and Swarm Intelligence Algorithms*, 2019, pp. 11-23.

[24] N. K. Jain, U. Nangia, and J. Jain, "A review of particle swarm optimization," *Journal of The Institution of Engineers (India): Series B,* vol. 99, pp. 407-411, 2018 2018, doi: 10.1016/j.cherd.2018.03.031.

[25] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," *IEEE Access,* vol. 10, pp. 10031-10061, 2022 2022, doi: 10.1109/ACCESS.2022.3142859.

[26] J. Cai, H. Wei, H. Yang, and X. Zhao, "A novel clustering algorithm based on DPC and PSO," *IEEE Access,* vol. 8, pp. 88200-88214, 2020 2020, doi: 10.1109/ACCESS.2020.2992903.

[27] M. Javani, K. Faez, and D. Aghlmandi, "Clustering and feature selection via PSO algorithm," in *2011 International Symposium on Artificial Intelligence and Signal Processing (AISP)*, 2011 2011, pp. 71-76, doi: 10.1109/AISP.2011.5960988.