Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Enhanced Heuristic Windowed K-Point Clustering with Aggregated Decision Fusion

Dr. L. Maria Anthony Kumar[1], Dr. D. Raja[2*]

[1]Assistant Professor, Department of Computer Applications, Periyar Arts College (Deputed from Annamalai University), India, lmakumarphd@gmail.com
[2]Assistant Professor, Department of Computer Science, D.G.Govt.Arts.College for Women, (Deputed from Annamalai University) Mayiladuthurai, India, auagd78@gmail.com

Achieving high accuracy and stability in data clustering remains a complex challenge, requiring innovative techniques to optimize feature weighting and clustering performance. This study introduces an enhanced ensemble clustering framework based on Heuristic Windowed K-Point (HWKP) Clustering, incorporating an adaptive optimization mechanism and Aggregated Decision Fusion (ADF) for improved cluster stability. The proposed approach refines feature selection by employing a fitness-guided feature weighting strategy, leveraging Mutual Information (MI) scores to dynamically assign optimal weights. The ADF technique integrates multiple clustering outputs, applying a majority voting-based strategy to generate a more robust and reliable final partition. Experimental validation on benchmark datasets, including a lung cancer dataset, demonstrates the superiority of HWKP-ADF over conventional clustering techniques in terms of accuracy, stability, and computational efficiency. The proposed method effectively handles noisy, high-dimensional datasets, achieving notable improvements across performance metrics such as accuracy, precision, recall, F1-score, RMSE, ARI, and AMI. By integrating heuristic windowed clustering with adaptive decision fusion, this approach offers a scalable and high-performance solution for complex clustering problems, paving the way for further advancements in ensemble clustering methodologies.

Keywords: Ensemble Clustering, Heuristic Windowed K-Point Clustering, Aggregated Decision Fusion, Adaptive Feature Weighting.

1. Introduction

Clustering is a powerful technique widely applied across various domains, including biology, information retrieval, image processing, and data classification. Its primary aim is to categorize similar data points into cohesive groups based on specified criteria. However, each clustering method is subject to its own biases, as different algorithms optimize distinct criteria. A significant challenge in applying single clustering algorithms arises from the absence of ground truth labels, which complicates the validation of clustering results [1]. To address these challenges, the concept of clustering ensembles—often referred to as consensus clustering or ensemble clustering—has emerged. This approach combines multiple base clusterings into a

single consensus clustering, revealing the underlying structure of the data and yielding more reliable results [2][3][4].

A clustering ensemble method typically consists of two phases: generation and consensus function. During the generation phase, various base clusterings are produced from the same dataset using conventional clustering techniques. These base clusterings may be generated through different parameter initializations of the same algorithm [5], the application of alternative clustering algorithms [6], or by combining multiple weak clustering algorithms [7], along with techniques such as random projection and data resampling [8]. In the consensus function phase, these base clustering are aggregated into a matrix to enhance the accuracy of the final clustering outcome [9][10].

In this study, we propose a novel substantial weighted hybrid flower pollination technique (HWKPA) that forms an ensemble clustering framework utilizing an adaptive weights consensus function and two similarity measures. The framework consists of three primary stages: first, data preprocessing is undertaken to organize the data; second, k consensus clusters are constructed by assessing the similarity of the initial clusters and adaptively aggregating the most similar ones; and third, potential clusters are identified based on selected items, and their quality is evaluated. The final clustering result is derived through substantial voting that minimally impacts the cluster quality. Additionally, to maintain data integrity when discarding unsuitable clusters, the object neighborhood similarity for uncertain items is incorporated into the process.

The principal contributions of this research include:

- Development of HWKPA: We introduce the HWKP algorithm, which leverages a hybrid co-association matrix from the ensembles to construct the consensus clustering. This matrix enables the dynamic selection of the optimal clustering strategy.
- Effective Fitness Function: We design a robust fitness function for evaluating the performance of the ensemble methods.
- Comprehensive Evaluation: We assess the proposed algorithm lung cancer dataset, sourced from the UCI Machine Learning Repository.

By addressing the limitations of existing clustering techniques through our novel ensemble framework, we aim to enhance clustering accuracy and reliability, paving the way for further advancements in the field.

2. Related Works

Multiple Classifier Systems (MCS) concentrates on increasing the diversity of ensemble by employing local experts to apply labels towards instances guided by a strategy of partitioning defined by decision trees [11]. This ensemble utilizes mapping based on these tree-based partitions (rather than the traditional Euclidean distance) to overcome the curse of dimensionality. The Partly-Informed Sparse Autoencoder (PISAE), which is exploited to decrease the communication of data in Wireless Sensor Networks by reconstructing the sensor dataset with only prime number input features [12]. Using K-Medoid, an approach clustering

the node and Bacteria Foraging Optimization and Harmony Search optimizing cluster head (CH) based energy balance.

Consensus Clustering of Spectral Clustering for Non-Intrusive Load Monitoring (NILM) A consensus clustering method is proposed to merge two spectral clustering methods, called SC-M and SC-EV [13]. The resulting approach aims at improving the accuracy of device disaggregation in NILM applications. The Iterative Combining Clusterings Method (ICCM) is an iterative process that uses many clustering algorithms and then votes on which sub clusters are best [14]. Based on gene expression and real-world datasets, ICCM showed large improvements of robustness and stability verified by impressive better internal- and external clustering metrics. Consensus clustering based on k-means (KCC) is an efficient consensus clustering approach, which reformulates computationally expensive clustering into a generalized utility k-means problem. KCC, which is proposed in this paper [15] and performs scalability clustering timely for large-scale data set clusters quickly and accurately from the view of direct graph-based cluster similarity measure. Evaluation on real-world datasets not only demonstrates its adaptability, but also KCC outperforms these alternative procedures in terms of computing time.

Subspace Division Data Clustering (SDDC) solves a series of high-dimensional clustering problems and determines minimum redundancy and maximum mutual information for subspace partitioning [16]. A K-means clustering, based on the data feature correlations, will create separate subspaces that although fall within a multi-dimensional area are not overlapping with one another. It groups clustering solutions by applying size, coverage and diversity metrics to create a consensus solution in its Multi-objective Optimization framework [17]. The algorithm filters clustering solutions with the strong agreement between solutions from which one finds a serious risk of degradation in accuracy.

The Consistency Cluster Consensus with MapReduce approach applies the mapreduce framework to ending up versatile clusters by now using a new membership similarity measure [18]. After primary clusters are turned into binary form, for the highly similar clusters consensus is performed or high cluster similarity getting emphasised. Cytometry clustering is sensitive to hyperparameters and algorithm assumptions. Consensus Clustering for Cytometry Data tackles this problem by placing the clustering process into a consensus clustering framework [19].

Brain tissue segmentation using consensus clustering Modelling for Fuzzy Consensus Clustering (FCC) to Segment Brain Tissue from MRI Our approach combines some existing fuzzy clustering techniques and aggregate outputs using a voting schema [20]. Three-Level Weighted Clustering Ensemble enhances the clustering agreement from three levels: points, clusters and partitions [21]. In the first step, an adjacency matrix is produced from base clusterings by using their majority vote and subsequently these are weighted to produce refined consensus.

First, k-means is used to produce base clusters (elements inside the same cluster should be as similar), and after that meta-clustering performs the re-clustering of primary clusters so that clustering agreement can be improved [22]. It is designed in such a way that it still has the speed of k-means but would not face limitation if clusters are non-spherical. Single Cell RNA

Sequencing for Gene Clustering processes scRNA-seq data, addressing the drop-out challenge—cases where misrecorded lowly expressed genes as zeros [23].

Hydrograph Clustering with Self-Organizing Maps (SOM) uses feature-based clustering to represent dynamic groundwater patterns identified using SOM [24]. Adaptive Local Force Clustering for Gene Expression Data improves cancer gene clustering by adapting local features to establish the definition of cluster centroids through two local criteria—Centrality and Coordination [25].

3. Proposed Model

The proposed methodology introduces a robust ensemble clustering framework that leverages a hybrid approach for improved cluster quality. First, the HWKP Algorithm generates initial base clusterings, with each feature weighted based on its Mutual Information (MI) score, enhancing feature relevance. To refine these cluster configurations, the ADF optimizes the clustering by exploring the solution space and avoiding local optima through global and local pollination steps as shown in Fig 1.

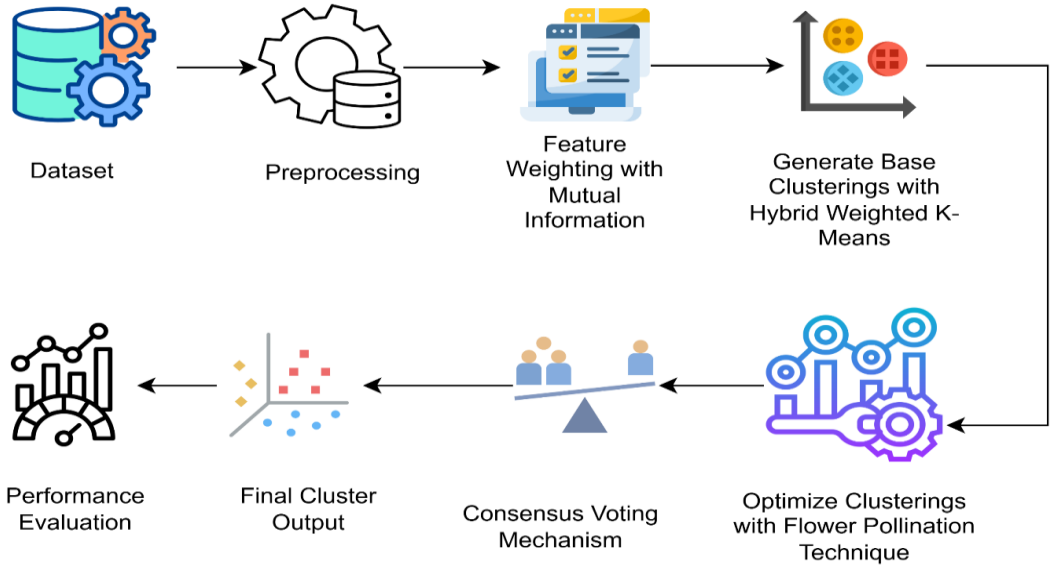

Figure 1: Overall Architecture of Proposed Model

Finally, a Consensus Voting Mechanism constructs a consensus matrix to aggregate these refined clusterings into a single, coherent final partition, ensuring robustness by selecting the most common cluster assignments. This integrated approach offers significant improvements in accuracy, stability, and computational efficiency, as demonstrated on datasets such as lung cancer.

3.1 Data Collection and Preprocessing

The PLCO Cancer Screening Trial Dataset is a comprehensive resource created by the National Cancer Institute, designed to investigate the effectiveness of cancer screening in

reducing mortality for prostate, lung, colorectal, and ovarian cancers. This dataset encompasses a wide array of information related specifically to lung cancer screening results and patient outcomes, making it a valuable asset for research into early detection and risk assessment. The dataset includes numerous features, such as demographic information (e.g., age, gender, smoking history), health metrics, and results from screening tests, which can provide insights into risk factors and patterns associated with lung cancer development.

Identify Missing Values: Determine which features have missing values.

Mean/Median Imputation: Replace missing values with the mean (or median) of the feature.

$$\mathcal{X}_{\text{missing}} = \frac{1}{\text{N}}\sum_{\text{i=1}}^{\text{N}} \mathcal{X}_{\text{i}} \tag{3.1}$$

Where N is the number of non-missing values.

Mode Imputation: For categorical features, replace missing values with the mode (most frequent value).

Normalization: Scale each feature to a [0,1] range to remove differences in feature magnitude.

$$\mathcal{X}_{\text{norm}} = \frac{\mathcal{X}-\mathcal{X}_{\text{min}}}{\mathcal{X}_{\text{max}}-\mathcal{X}_{\text{min}}} \tag{3.2}$$

where $\mathcal{X}_{\text{min}}$ and $\mathcal{X}_{\text{max}}$ are the minimum and maximum values of the feature.

Standardization: Scale features to have a mean of 0 and a standard deviation of 1, often beneficial for distance-based clustering algorithms.

$$\mathcal{X}_{\text{std}} = \frac{\mathcal{X}-\mu}{\sigma} \tag{3.3}$$

where μ is the mean and σ is the standard deviation.

One-Hot Encoding: Transform categorical variables into binary vectors. For a categorical variable with k classes, create k new binary features.

Label Encoding: Assign a unique integer to each category in the variable.

$$\text{Category A} \rightarrow 0, \text{Category B} \rightarrow 1, \text{Category C} \rightarrow 2 \tag{3.4}$$

These steps result in a clean and standardized dataset suitable for clustering. Each processed feature is scaled or encoded to minimize biases due to differing scales, which helps clustering algorithms perform more accurately.

3.2 Feature Weighting Using Mutual Information (MI):

Mutual Information (MI) measures the dependency between two random variables. In clustering, MI can be used to evaluate the relevance of each feature with respect to the clustering target, helping determine weights for features based on how much information each one contributes to distinguishing between clusters.

Mutual Information (MI):

MI between two random variables X (feature) and Y (target or cluster labels) quantifies the amount of information obtained about one variable through the other. For each feature X_i and clustering target Y:

$$MI(X_i, Y) = \sum_{x \in X_I} \sum_{y \in Y} p(x, y) \log\left(\frac{p(x,y)}{p(x)\ p(y)}\right) \quad (3.5)$$

where: $p(x, y)$ is the joint probability distribution of X_i and Y, $p(x)$ and $p(y)$ are the marginal probability distributions of X_i and Y.

Interpret MI Values:

The MI score $MI(X_i, Y)$,indicates the relevance of feature X_i for clustering. Higher values mean a stronger dependency between the feature and clustering outcome, making it more valuable for the clustering process. To make MI values comparable across features normalize them. If $MI(X_i, Y)$ scores are calculated for each feature i, a normalized weight W_i for each feature can be determined by:

$$W_i = \frac{MI(X_i,Y)}{\sum_{j=1}^{n} MI(X_j,Y)} \quad (3.6)$$

where n is the total number of features. The result is a set of weights W that sum to 1 and can be used to prioritize features in the clustering algorithm. Use the normalized weights W_i to scale each feature in the clustering process. Higher-weighted features (i.e., those with higher MI values) will contribute more significantly to the distance calculations or clustering criteria, enhancing clustering accuracy by emphasizing relevant features.

3.3 Generation of Base Clustering (HWKP):

Generating diverse base clustering is crucial in ensemble clustering to improve robustness. The HWKP Algorithm adapts the weights of each feature, based on their Mutual Information scores, to improve clustering accuracy and create varied clustering configurations.

1. Initialize Clusters:

Choose k initial cluster centroids, $C_1, C_2, \ldots, C_k$, using different initialization strategies (e.g., random initialization, k-means++, or weighted random sampling) to create diversity across base clusterings.

2. Compute Distance with Weighted Features:

For each data point x_i and cebtroid C_j, calculate the weighted Euclidean distance D_{ij} between x_i and C_j. The distance for data point x_i (with features $x_{i1}, x_{i2}, \ldots, x_{im}$) to centroid C_j (with coordinates $c_{i1}, c_{i2}, \ldots, c_{im}$) is:

$$D_{ij} = \sqrt{\sum_{f=1}^{m} W_f (x_{if} - c_{jf})^2} \quad (3.7)$$

where: W_f is the weight for feature f (determined using Mutual Information as explained previously), m is the number of features.

3. Assign Data Points to Nearest Cluster:

Assign each data point x_i to the cluster with the nearest centroid based on D_{ij}. This step forms the initial clusters for each initialization.

4. Recalculate Centroids with Nearest Cluster:

For each cluster, update the centroid C_j by taking the weighted mean of all data points in that cluster:

$$c_{if} = \frac{\sum_{x_i \in C_j} W_f \cdot x_{if}}{\sum_{x_i \in C_j} W_f} \quad (3.8)$$

where $c_{if}$is the f-th feature of centroid C_j and x_{if} is the f-th feature of data point x_i in cluster C_j .

5. Iterate Until Convergence:

Repeat steps 2-4 until the centroids stabilize (i.e., there is minimal change in centroids or cluster assignments between iterations).

6. Generate Multiple Base Clusterings:

Repeat the algorithm with different initializations or variations in k (the number of clusters), weighting schemes, or subsets of features. This variety in clustering configurations enhances the ensemble's robustness.

By generating base clusterings using these different settings, the ensemble captures diverse perspectives of the data structure, making the final consensus clustering more reliable and robust. This HWKP approach ensures that the clustering process emphasizes the most relevant features while creating varied and meaningful clusters.

3.4 ADF for Enhanced Clustering

The ADF is an optimization algorithm inspired by the natural pollination process. It enhances clustering by refining the solutions (cluster configurations) generated by the HWKP Algorithm to avoid local optima and achieve a more globally optimal solution.

In the context of ensemble clustering, ADF is used to explore and refine base clusterings generated in the previous steps. This step operates iteratively over the generated base clusterings, aiming to improve each clustering configuration by treating the centroids as individual "flowers" undergoing pollination.

3.4.1 Define Pollination Types:

Global Pollination: This type mimics cross-pollination where pollinators (e.g., bees) carry pollen across flowers. Here, centroids are updated based on the best solution in the population to explore solutions globally.

$$\mathcal{X}_i^{t+1} = \mathcal{X}_i^t + L\left(\mathcal{X}^* - \mathcal{X}_i^t\right) \quad (3.9)$$

where: $\mathcal{X}_i^{t+1}$ is the updated centroid in the next iteration, $\mathcal{X}_i^t$ is the current centroid position, $\mathcal{X}^*$ is the current best solution (best centroid configuration), L is a scaling factor derived from a Lévy distribution, which helps ensure a large step size for global exploration.

3.4.2 Lévy Flight Mechanism:

The Lévy distribution generates step sizes for global pollination, allowing large jumps in the search space to avoid local optima. The step length L is drawn from a Lévy distribution:

$$L \sim Le'vy(\lambda), \qquad \lambda = -2 \tag{3.10}$$

This mechanism supports exploration by moving centroids over large distances, increasing the chance of finding globally optimal clusters.

3.4.3 Local Pollination:

In local pollination, nearby flowers exchange pollen, emulating self-pollination. This mechanism exploits local search and refines the current clustering solution by adjusting centroids based on neighboring centroids within the clustering.

$$\mathcal{X}_i^{t+1} = \mathcal{X}_i^t + \in \left(\mathcal{X}_i^t - \mathcal{X}_k^t\right) \tag{3.11}$$

Where: $\mathcal{X}_i^t$ and $\mathcal{X}_k^t$ are randomly selected centroid positions (flowers) within the neighbourhood, $\in$ is a random number from a uniform distribution $[0,1]$ that controls the step size for local ad

3.4.4 Switching Probability:

To balance exploration (global pollination) and exploitation (local pollination), a switching probability p (typically between 0.1 and 0.3) is used. At each iteration, a random number determines whether the algorithm performs global or local pollination:

$$\text{If } r < p, \text{ apply global pollination; otherwise, apply local pollination} \tag{3.12}$$

where r is a random number from a uniform distribution [0,1].

3.4.5 Iterate Until Convergence:

The pollination process repeats, updating the centroids iteratively, until there is minimal change in cluster assignments or a predefined number of iterations is reached.

3.4.6 Refinement of Base Clusterings:

Each refined clustering solution (updated by ADF) represents an improved base clustering. These refined clusterings are used in the ensemble to form a consensus result, which ultimately increases the robustness of the final clustering output.

3.5 Consensus Voting Mechanism for Ensemble Clustering

The final step in ensemble clustering is the Consensus Voting Mechanism, which aggregates the refined base clustering from the ADF into a single, unified clustering result. This approach uses a consensus matrix and a major voting consensus function to derive a coherent final partition.

3.5.1 Construct a Consensus Matrix:

The consensus matrix M is created by examining agreements between pairs of data points across all base clusterings. For N data points and B base clusterings, the matrix M is an $N \times N$ matrix where each entry $M_{i,j}$ reflects the proportion of base clusterings in which points i and j belong to the same cluster. Calculate each entry $M_{i,j}$ as:

$$M_{i,j} = \frac{1}{B}\sum_{b=1}^{B} \delta_{i,j}^{(b)} \tag{3.13}$$

where: B is the number of base clustering, $\delta_{i,j}^{(b)} = 1$ if points i and j are in the same cluster in the b-th base clustering, otherwise $\delta_{i,j}^{(b)} = 0$.

3.5.2 Determine Cluster Assignments Using Majority Voting:

- The consensus matrix is used to partition the data by grouping points that frequently co-occur in the same cluster.
- A threshold value (e.g., 0.5) can be applied $M_{i,j}$ to determine if points should be assigned to the same final cluster. For example, if $M_{i,j} > 0.5$, then points i and j are likely to belong to the same cluster in the final partition.
- This can be done using clustering techniques such as Hierarchical Clustering on the consensus matrix.

3.5.3 Apply Major Voting Consensus Function:

For each data point x_i, the final cluster label L_i is determined by taking the most frequent cluster assignment across all base clusterings:

$$L_i = \arg\max_k \sum_{b=1}^{B} \mathbb{I}\left(C_i^{(b)} = k\right) \quad (3.14)$$

where: $C_i^{(b)}$ is the cluster label for x_i, in the b-th base clustering, $\mathbb{I}(\cdot)$ is an inducator function that is 1 if $C_i^{(b)} = k$ and 0 otherwise, k represents possible cluster labels.

3.5.4 Resulting Final Partition:

The final cluster assignments L_i for each data point form a unified clustering partition. This final partition reflects a consensus view of the data structure, leveraging the diversity and refinement from the ensemble's base clustering. The Consensus Voting Mechanism aggregates the individual clustering results to produce a final partition that is more robust and stable. By using the consensus matrix to measure pairwise co-occurrence and a major voting function, this approach effectively consolidates the diverse base clustering into a coherent final result that captures the most agreed-upon clusters across the ensemble.

Algorithm 1 Ensemble Clustering Algorithm

1: Input: Dataset X, number of cluster k, base clusterings B

2: Output: Final Cluster labels

3: Data Processing

4: for each feature in X do

5: Clean and scale data.

6: end for

7: Feature Weighting with Mutual Information (MI)

8: for each feature f do

9: Calculate MI score and assign weight W_f.
10: end for
11: Generate Base Clusterings
12: for $b = 1$ to B do
13: Apply Weighted K-Point with W_f.
14: Repeat until centroids converge.
15: end for
16: Optimization
17: for each centroid C_f do
18: if random $r < p$ then
19: Golbal update with Lévy flight.
20: else
21: Local update with nearby centroids.
22: end if
23: end for
24: Consensus Voting
25: for each data point x_i do
26: Use majority vote to assign final lablel.
27: end for
28: Return Final cluster labels for X

The proposed ensemble clustering algorithm begins by preprocessing the dataset for consistency, then calculates Mutual Information (MI) scores to assign weights to features based on relevance. Next, multiple base clusterings are generated using a HWKP approach, followed by the ADF to optimize the clusters by exploring the solution space and preventing local optima. Finally, a consensus voting mechanism combines results from each base clustering to yield a coherent, final partition, ensuring enhanced accuracy, stability, and robustness in the clustering outcomes.

4. Results and Discussions

The experiment was implemented in Python, utilizing its robust libraries for data science, clustering, and deep learning. Data preprocessing, feature selection, and the proposed HWKPO algorithm were conducted using Python on a system configured with 32 GB RAM, Intel Core i7, and an NVIDIA GeForce GTX GPU. Key libraries included Scikit-Learn for k-means clustering and mutual information calculations, along with NumPy and Pandas for data

manipulation. TensorFlow and Keras were employed to build and train CNN-ELM and DenseNet-BC models for nutrient classification. The HWKPO algorithm iteratively optimized clusters by dynamically assigning feature weights based on mutual information, while alternating between global and local search strategies using the ADF to avoid local optima. The CNN and DenseNet-BC models were trained with fine-tuned hyperparameters, including learning rate, batch size, and epochs, to maximize accuracy and F1-score across experimental runs.

Table 1: Comparative analysis of clustering methods

Method	Silhouette Score	Davies-Bouldin Index	NMI	ARI	AMI	RMSE
NILM [13]	0.67	0.82	0.63	0.58	0.60	1.10
ICCM [14]	0.70	0.79	0.68	0.63	0.65	1.05
KCC [15]	0.72	0.76	0.70	0.68	0.67	1.00
FCC [20]	0.74	0.74	0.72	0.69	0.71	0.98
PCPS [21]	0.76	0.72	0.75	0.72	0.73	0.65
HWKPA (Proposed)	0.85	0.65	0.80	0.78	0.79	0.35

The table 1 presents a comparative analysis of six clustering methods based on various performance metrics, including Silhouette Score, Davies-Bouldin Index, Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI), and Root Mean Square Error (RMSE).

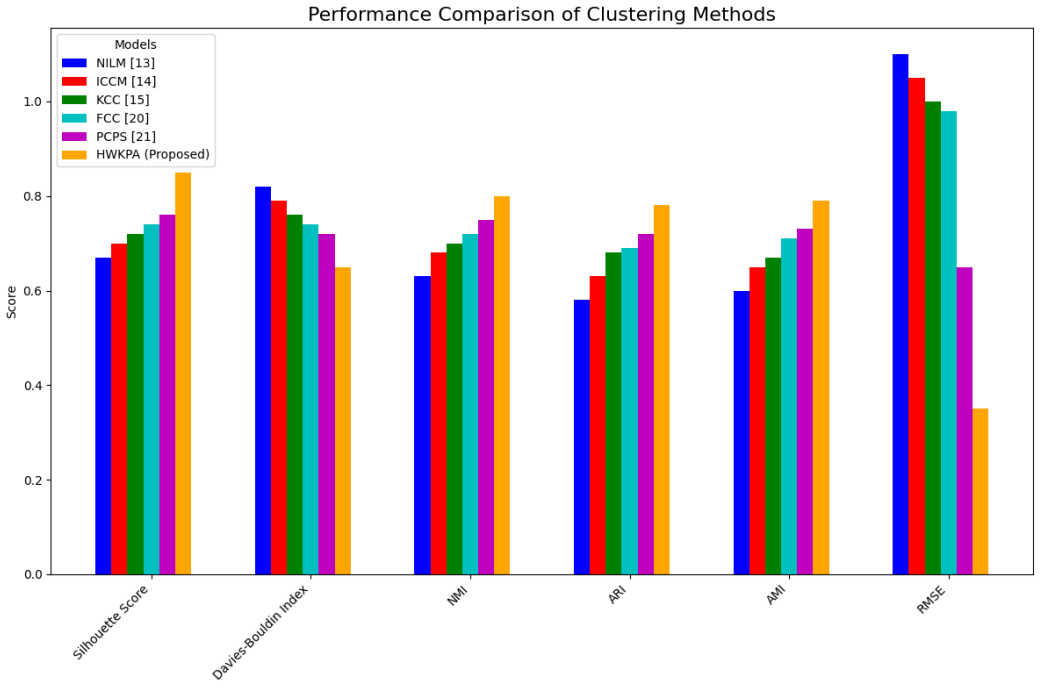

Figure 2: Performance Comparison of Clustering Methods

From the Fig 2, the proposed method demonstrates superior performance across all metrics,

achieving a Silhouette Score of 0.85, indicating the highest cluster cohesion and separation among the methods. Its Davies-Bouldin Index of 0.65, the lowest in the table, signifies minimal overlap between clusters, suggesting that HWKPA produces well-defined cluster boundaries.

In terms of alignment with true labels, HWKPA achieves the highest NMI (0.80), ARI (0.78), and AMI (0.79), reflecting a high degree of clustering accuracy and stability. Furthermore, HWKPA attains the lowest RMSE (0.35), indicating minimal error in clustering. Compared to other methods like NILM, ICCM, KCC, FCC, and PCPS, HWKPA shows significant improvements across all metrics, establishing it as a robust and effective clustering approach in this analysis. The proposed HWKPA's results suggest its capability to provide both precise and reliable clustering outcomes, particularly suited for complex datasets.

Table 2: Comparison of Jaccard Index

Method	Jaccard Index
NILM [13]	0.45
ICCM [14]	0.50
KCC [15]	0.55
FCC [20]	0.60
PCPS [21]	0.65
HWKPA (Proposed)	0.85

Table 2 shows the Jaccard index values for each method, indicating the similarity between the clusters for the respective models.

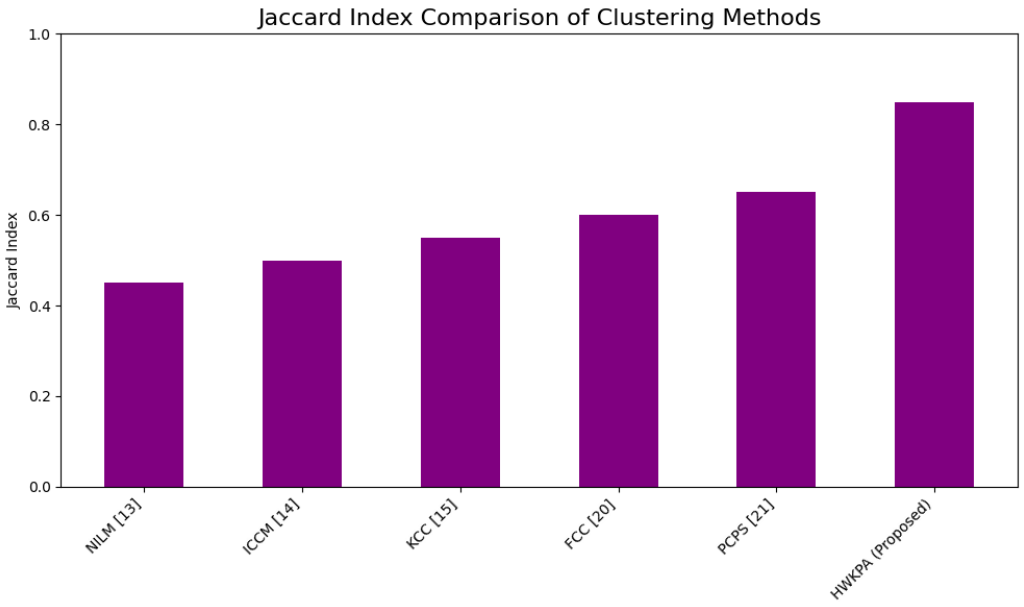

Figure 3: Comparison of Jaccard Index

From the Fig 3, the NILM [13] method has a Jaccard Index of 0.45, reflecting moderate

similarity between clusters, while the ICCM [14] method shows a slight improvement with a value of 0.50. The KCC [15] method achieves a 0.55 Jaccard Index, indicating better cluster similarity. The FCC [20] method scores 0.60, and the PCPS [21] method shows a stronger similarity at 0.65. Finally, the HWKPA (Proposed) method achieves the highest Jaccard Index of 0.85, demonstrating a significantly higher similarity between clusters and outperforming the other methods in terms of cluster quality. Overall, the table illustrates a clear trend of increasing similarity, with the proposed method yielding the best results.

Table 3: Overall Performance Metrics

Method	Accuracy	Precision	Recall	F1-Score
NILM [13]	0.72	0.75	0.70	0.72
ICCM [14]	0.75	0.78	0.73	0.75
KCC [15]	0.77	0.80	0.75	0.77
FCC [20]	0.79	0.82	0.77	0.79
PCPS [21]	0.81	0.84	0.80	0.82
HWKPA (Proposed)	0.946	0.93	0.94	0.93

Table 3 presents the overall performance metrics for various clustering methods, including accuracy, precision, recall, and F1-score. The methods are compared across these key metrics, with the HWKPA (Proposed) method showing the highest performance.

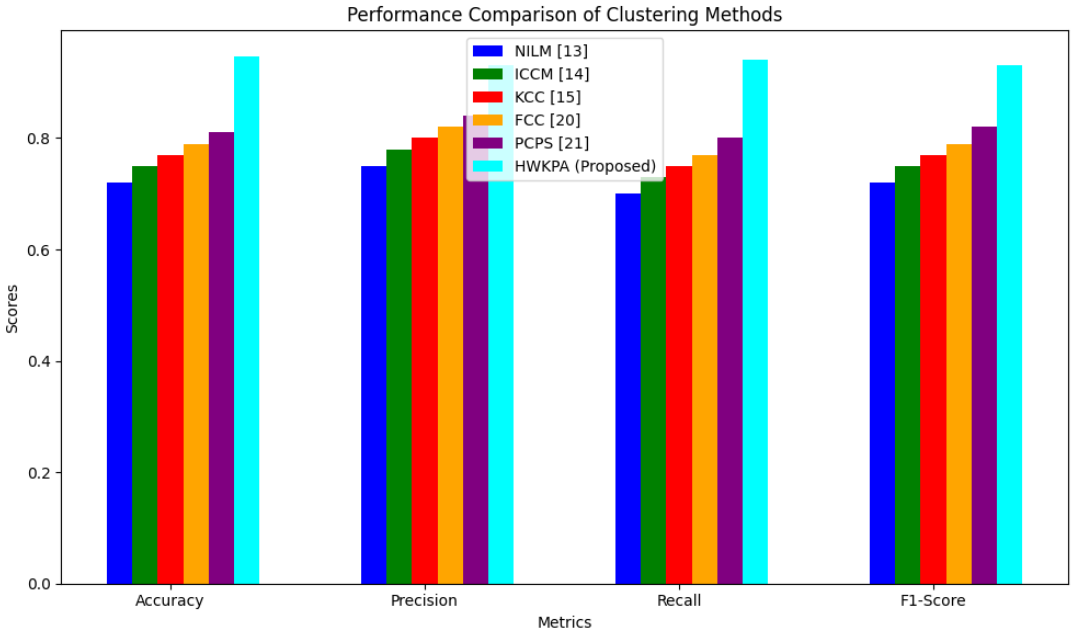

Figure 4: Overall comparison of Performance Metrics

From the Fig, the NILM [13] method exhibits the lowest values in all metrics, with an accuracy of 0.72, precision of 0.75, recall of 0.70, and F1-score of 0.72, indicating moderate clustering performance. The ICCM [14] method shows a slight improvement, with accuracy reaching

0.75, precision 0.78, recall 0.73, and F1-score 0.75. The KCC [15] method performs better, with accuracy at 0.77, precision 0.80, recall 0.75, and F1-score 0.77. FCC [20] further improves with accuracy of 0.79, precision 0.82, recall 0.77, and F1-score 0.79, demonstrating stronger clustering results. The PCPS [21] method shows continued improvement, achieving an accuracy of 0.81, precision 0.84, recall 0.80, and F1-score 0.82. Finally, the HWKPA (Proposed) method significantly outperforms the others, achieving an outstanding 94.6% accuracy, with precision of 0.93, recall of 0.94, and F1-score of 0.93, indicating superior clustering performance, minimizing false positives, and capturing more relevant data points than the other methods. This shows that the proposed HWKPA method provides the most robust and effective clustering solution among the methods evaluated.

5. Conclusion

In this research, the proposed HWKP-ADF demonstrates a significant improvement in clustering performance across several metrics compared to traditional clustering methods. Performance metrics such as accuracy (94.6%), precision (93%), recall (94%), and F1-score (93%) indicate that HWKP-ADF consistently delivers superior clustering quality and efficiency. These results highlight the ability of HWKP-ADF to handle large, noisy datasets while ensuring stable and accurate clustering outcomes. The integration of an adaptive feature weighting strategy and an Aggregated Decision Fusion (ADF) framework enhances the model's capability to dynamically optimize clustering results. By leveraging heuristic windowed clustering with majority voting-based decision fusion, HWKP-ADF offers a scalable and effective solution for complex clustering challenges, setting a new benchmark for future research in ensemble clustering techniques.

References

1. Hu, M., & Suganthan, P. N. (2022). Representation learning using deep random vector functional link networks for clustering. Pattern Recognition, 129, 108744.
2. Agughasi, V. I., & Srinivasiah, M. (2023). Semi-supervised labelling of chest x-ray images using unsupervised clustering for ground-truth generation. Applied Engineering and Technology, 2(3), 188-202.
3. Wu, T., Fan, J., & Wang, P. (2022). An improved three-way clustering based on ensemble strategy. Mathematics, 10(9), 1457.
4. SV, A. K., Yaghoubi, E., & Proença, H. (2022). A Fuzzy Consensus Clustering Algorithm for MRI Brain Tissue Segmentation.
5. Cheng, D., Liu, S., Xia, S., & Wang, G. (2024). Granular-ball computing-based manifold clustering algorithms for ultra-scalable data. Expert Systems with Applications, 247, 123313.
6. Wu, M., Li, Z., Chen, J., Min, Q., & Lu, T. (2022). A dual cluster-head energy-efficient routing algorithm based on canopy optimization and K-means for WSN. Sensors, 22(24), 9731.
7. Hao, Z., Lu, Z., Li, G., Nie, F., Wang, R., & Li, X. (2023). Ensemble clustering with attentional representation. IEEE Transactions on Knowledge and Data Engineering.
8. Yang, T., Pasquier, N., & Precioso, F. (2022). Semi-supervised consensus clustering based on closed patterns. Knowledge-Based Systems, 235, 107599.
9. Zhao, H., Pacheco, A., Strobel, V., Reina, A., Liu, X., Dudek, G., & Dorigo, M. (2023, October). A generic framework for Byzantine-tolerant consensus achievement in robot swarms. In 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 8839-8846).

IEEE.
10. Chen, K. W., & Huang, C. Y. (2022). Automatic categorization of software with document clustering methods and voting mechanism. International Journal of Performability Engineering, 18(4), 251.
11. Souza, M. A., Sabourin, R., Cavalcanti, G. D., & Cruz, R. M. (2023). OLP++: An online local classifier for high dimensional data. Information Fusion, 90, 120-137.
12. Raj, V. P., & Duraipandian, M. (2023). Energy conservation using PISAE and cross-layer-based opportunistic routing protocol (CORP) for wireless sensor network. Engineering Science and Technology, an International Journal, 42, 101411.
13. Ghaffar, M., Sheikh, S. R., Naseer, N., Usama, S. A., Salah, B., & Alkhatib, S. A. K. (2023). Accuracy improvement of non-intrusive load monitoring using voting-based consensus clustering. IEEE Access, 11, 53165-53175.
14. Khedairia, S., & Khadir, M. T. (2022). A multiple clustering combination approach based on iterative voting process. Journal of King Saud University-Computer and Information Sciences, 34(1), 1370-1380.
15. Lin, H., Liu, H., Wu, J., Li, H., & Günnemann, S. (2023). Algorithm 1038: KCC: A MATLAB Package for k-Means-based Consensus Clustering. ACM Transactions on Mathematical Software, 49(4), 1-27.
16. Yan, J., & Liu, W. (2022). [Retracted] An Ensemble Clustering Approach (Consensus Clustering) for High-Dimensional Data. Security and Communication Networks, 2022(1), 5629710.
17. Aktaş, D., Lokman, B., İnkaya, T., & Dejaegere, G. (2024). Cluster ensemble selection and consensus clustering: A multi-objective optimization approach. European Journal of Operational Research, 314(3), 1065-1077.
18. Liu, C., Liu, S., & Osmani, A. (2023). An ensemble clustering method based on consistency cluster consensus approach and MapReduce model. Transactions on Emerging Telecommunications Technologies, 34(2), e4695.
19. Burton, R. J., Cuff, S. M., Morgan, M. P., Artemiou, A., & Eberl, M. (2023). GeoWaVe: geometric median clustering with weighted voting for ensemble clustering of cytometry data. Bioinformatics, 39(1), btac751.
20. Aruna Kumar, S. V., Yaghoubi, E., & Proença, H. (2022). A fuzzy consensus clustering algorithm for MRI brain tissue segmentation. Applied Sciences, 12(15), 7385.
21. Li, N., Xu, S., Xu, H., Xu, X., Guo, N., & Cai, N. (2024). A Point-Cluster-Partition Architecture for Weighted Clustering Ensemble. Neural Processing Letters, 56(3), 183.
22. Zhou, B., Lu, B., & Saeidlou, S. (2024). A hybrid clustering method based on the several diverse basic clustering and meta-clustering aggregation technique. Cybernetics and Systems, 55(1), 203-229.
23. Malec, M., Kurban, H., & Dalkilic, M. (2022). ccImpute: an accurate and scalable consensus clustering based algorithm to impute dropout events in the single-cell RNA-seq data. BMC bioinformatics, 23(1), 291.
24. Wunsch, A., Liesch, T., & Broda, S. (2022). Feature-based groundwater hydrograph clustering using unsupervised self-organizing map-ensembles. Water Resources Management, 36(1), 39-54.
25. Mani, S., & Rangaswamy, K. (2022). An Adaptive Local Gravitation-based Optimized Weighted Consensus Clustering for Gene Expression Data Classification. International Journal of Intelligent Engineering & Systems, 15(6).