DOI Link: https://doi.org/10.61586/zTphv
Vol.40, Issue.6, Part.1, June 2025, PP. 2-22
Received April 2025 Accepted May 2025 Published June 2025

PERFORMANCE ANALYSIS OF MULTI-SWARM AI-BASED PSO, DE, AND K-MEANS FOR OPTIMIZATION TASKS

Sitanath Biswas[1*], Dr. V Senthil Kumar[2], Dr A Ravi[3], Dr. Divya Kumari Tankala[4], Bhupchand Kumhar[5], Sayan Chakraborty[6], Shubhashree Sahoo[7]

[1, 6, 7] JIS College of Engineering, Kalyani, West Bengal, India

2Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, INDIA

[3]Moulana Azad National Urdhu university, Hyderabad, INDIA

[4]Department of CSE, G. Narayanamma Institute of Technology and Science, Hyderabad, INDIA

[5]Department of Computer Science and Engineering, IES Institute of Technology and Management, IES University, Bhopal, Madhya Pradesh, India

**Corresponding Author*: *sitanath.biswas@jiscollege.ac.in*

Abstract

This study evaluates a multi-swarm AI framework that embeds K-Means within Particle Swarm Optimization (PSO) and Differential Evolution (DE) to overcome classic clustering weaknesses. Across benchmark datasets, DE delivers the best balance of accuracy and speed: Silhouette = 0.67, Davies-Bouldin = 0.69, MSE = 0.045, RMSE = 0.212, and quicker convergence than PSO. PSO improves on plain K-Means (Silhouette = 0.61 vs 0.52; DBI = 0.75 vs 0.86; MSE = 0.052 vs 0.078) but converges slower than DE because of velocity-update dependence. K-Means remains the fastest computationally yet lags markedly in clustering quality. Overall, multi-swarm optimization boosts clustering performance but increases computational demands and complicates hyper-parameter tuning. Future work should devise adaptive mechanisms to trim runtime and automate parameter selection to broaden real-world applicability.

Keywords - Differential Evolution (DE), K-Means, Silhouette Score, Davies-Bouldin Index (DBI), Multi-swarm optimization.

1. Introduction

Optimization algorithms play a crucial role in addressing complex real-world problems across various domains, including machine learning, data clustering, engineering design, and financial modeling. As computational challenges grow in complexity, there is a need for robust, adaptive, and scalable optimization techniques capable of efficiently exploring large solution spaces. Metaheuristic algorithms, inspired by natural and biological processes, have emerged as powerful tools for handling global optimization problems, particularly in high-dimensional, nonlinear, and multimodal landscapes (Kennedy & Eberhart, 1995; Storn & Price, 1997). Among the widely used metaheuristic methods, Particle Swarm Optimization (PSO), Differential Evolution (DE), and K-Means clustering are particularly notable. PSO, introduced by Kennedy and Eberhart (1995), is a population-based optimization technique inspired by the collective intelligence observed in bird flocking and fish schooling. It has been successfully applied to various optimization tasks, but it often suffers from premature convergence due to a lack of exploration. DE, developed by Storn and Price (1997), is an evolutionary algorithm that employs mutation, crossover, and selection to iteratively refine solutions. It has demonstrated superior performance in continuous optimization problems but can be computationally intensive, especially for large-scale applications. K-Means clustering, first proposed by MacQueen (1967), is a centroid-based clustering method widely used in data analysis and pattern recognition. However, it is highly sensitive to initial centroid selection, which can lead to suboptimal clustering results (Likas et al., 2003). Despite their individual advantages, these algorithms have inherent limitations, such as local optima stagnation, parameter sensitivity, and inefficient handling of complex multimodal functions. To mitigate these issues, multi-swarm AI-based optimization techniques have been introduced, leveraging the cooperative behavior of multiple sub-populations to enhance search diversity and convergence speed (Tan et al., 2020). By integrating multiple swarms, these hybrid models offer a more balanced exploration-exploitation tradeoff, making them well-suited for high-dimensional and real-world optimization challenges. This study investigates the performance analysis of Multi-Swarm AI-based PSO, DE, and K-Means in various optimization contexts. The primary objectives are to evaluate their solution quality, convergence behavior, computational efficiency, and robustness across benchmark functions and real-world datasets. Through a comparative analysis, this research aims to provide insights into the suitability of these algorithms for different optimization applications and propose potential improvements for enhanced performance.

2. Literature Review

Particle Swarm Optimization (PSO), introduced by Kennedy and Eberhart (1995), is a population-based metaheuristic inspired by the collective behavior of birds and fish. Each particle in the swarm adjusts its trajectory based on personal experience and social interaction with neighboring particles, allowing for efficient search and optimization. However, traditional PSO is prone to premature convergence, especially in high-dimensional and multimodal search spaces (Eberhart & Shi, 2001). To address these limitations, various multi-swarm PSO strategies have been developed. These include Cooperative PSO (CPSO), where multiple subswarms independently explore different regions of the search space and periodically share information (Van den Bergh & Engelbrecht, 2004), and Dynamic Multi-Swarm PSO (DMS-PSO), which dynamically adjusts the number and distribution of swarms to maintain diversity (Liang & Suganthan, 2005). Studies have shown that multi-swarm PSO variants significantly improve solution accuracy and convergence speed, making them well-suited for large-scale optimization problems. Differential Evolution (DE), developed by Storn and Price (1997), is an evolutionary optimization technique known for its ability to handle complex global optimization tasks. It operates through mutation, crossover, and selection processes, allowing the population to evolve towards optimal solutions. Despite its robustness, DE can suffer from slow convergence and high computational costs in high-dimensional problems (Das et al., 2009). Several improvements have been proposed, including Self-Adaptive DE (SADE), which dynamically adjusts control parameters to enhance convergence speed (Brest et al., 2006), and Hybrid DE approaches, which integrate DE with other metaheuristics, such as PSO and Genetic Algorithms (GA), to improve performance (Zhang & Sanderson, 2009). These enhancements have demonstrated better balance between exploration and exploitation, reducing stagnation and improving convergence reliability.

K-Means clustering (MacQueen, 1967) is a widely used partitioning algorithm that minimizes intra-cluster variance by iteratively updating centroids. Despite its computational efficiency, K-Means suffers from sensitivity to initial centroid selection, leading to suboptimal clustering outcomes (Likas et al., 2003). To mitigate these issues, hybrid models have been proposed, such as PSO-K-Means and DE-K-Means, which use swarm intelligence to optimize centroid initialization and enhance clustering accuracy (Ma et al., 2016). Additionally, Fuzzy C-Means (FCM)-based hybrids have been explored to improve clustering robustness in noisy datasets (Bezdek et al., 1984). Studies indicate that swarm intelligence-based clustering methods outperform traditional K-Means in terms of clustering quality and convergence speed. Recent advancements in multi-swarm AI-based optimization have demonstrated superior performance in solving high-dimensional, real-world problems. Hybrid multi-swarm models have been applied in various domains, including bioinformatics (Li et al., 2021), image segmentation (Wang et al., 2020), and financial forecasting (Gao & Zhang, 2019). These studies highlight the benefits of integrating multiple swarm intelligence techniques to improve search efficiency, diversity maintenance, and global convergence.

Table-1: Summery of literature review

Name of the Author	**Methodology Used**	**Findings**	**Drawback**
Kennedy & Eberhart (1995)	Introduced Particle Swarm Optimization (PSO)	PSO effectively optimizes complex functions using swarm intelligence	Prone to premature convergence
Storn & Price (1997)	Developed Differential Evolution (DE)	Robust in solving continuous global optimization problems	Slow convergence in high-dimensional spaces
Van den Bergh & Engelbrecht (2004)	Proposed Cooperative PSO (CPSO)	Improved search diversity and solution accuracy	Increased computational overhead
Brest et al. (2006)	Developed Self-Adaptive DE (SADE)	Dynamically adjusts parameters for better convergence	Higher complexity in tuning parameters

Likas et al. (2003)	Studied K-Means clustering	Fast and efficient for large datasets	Sensitive to centroid initialization
Ma et al. (2016)	Hybrid PSO-K-Means clustering	Improved clustering accuracy and stability	Increased computational cost
Wang et al. (2020)	Multi-swarm AI-based segmentation	Enhanced image segmentation accuracy	Parameter tuning challenges
Li et al. (2021)	Multi-swarm models in bioinformatics	Improved feature selection and optimization	Higher computational demands

Despite extensive research on individual optimization algorithms and their hybrid variants, a significant gap remains in the comparative performance analysis and integration of Multi-Swarm AI-based Particle Swarm Optimization (PSO), Differential Evolution (DE), and K-Means clustering. While PSO and DE have been widely explored for various optimization tasks (Kennedy & Eberhart, 1995; Storn & Price, 1997), traditional approaches often struggle with premature convergence and inefficient exploration-exploitation balance (Eberhart & Shi, 2001; Brest et al., 2006). Multi-Swarm approaches have been introduced to enhance diversity (Van den Bergh & Engelbrecht, 2004), yet their effectiveness across diverse problem domains remains underexplored. Furthermore, existing hybrid models primarily focus on two-way integrations, such as PSO-K-Means and DE-K-Means (Ma et al., 2016; Likas et al., 2003), rather than a fully integrated Multi-Swarm AI-based framework that leverages the collective intelligence of all three algorithms. Additionally, while Multi-Swarm AI-based techniques have been applied to specific domains like image segmentation (Wang et al., 2020) and bioinformatics (Li et al., 2021), a broader comparative analysis across multiple benchmark datasets and real-world optimization problems is lacking. Another challenge lies in parameter tuning, as self-adaptive techniques have been proposed for PSO (Eberhart & Shi, 2001) and DE (Brest et al., 2006), but adaptive control strategies tailored for Multi-Swarm AI-based hybrid models remain underdeveloped (Das et al., 2009). Moreover, most studies evaluate optimization models on low- or medium-dimensional problems (Gao & Zhang, 2019), while research on a unified Multi-Swarm AI-based framework for high-dimensional and large-scale optimization is still insufficient. Addressing these gaps, this research aims to develop an integrated and dynamically adaptive Multi-Swarm AI-based optimization model that enhances solution quality, convergence speed, and computational efficiency across diverse applications. The proposed research focuses on developing a Multi-Swarm AI-based hybrid framework that integrates Particle Swarm Optimization (PSO), Differential Evolution (DE), and K-Means to enhance optimization performance. It includes a systematic performance evaluation on benchmark functions and real-world applications, alongside the implementation of an adaptive parameter control mechanism aimed at improving search efficiency and convergence behavior. To validate its effectiveness, the model will be compared against state-of-the-art optimization techniques, ensuring a comprehensive assessment of its capabilities in tackling complex optimization problems.

3. Methodologies

3.1. DATASET DESCRIPTION

The dataset, titled OYO_HOTEL_ROOMS, contains information about hotels listed on the OYO platform. It includes 791 entries, each representing an individual hotel, and 6 attributes: Hotel_name, Location, Price, Discount, and Rating. The Hotel_name column lists the names of the hotels, and the Location column specifies their geographical locations, typically including cities and areas within cities, such as Mumbai. The Price column indicates room prices, presumably in INR, and the Discount column shows the corresponding discount rates as decimal fractions. The Rating column provides a numeric score for the hotels, likely reflecting user reviews or other metrics. This dataset offers a comprehensive view of OYO hotels, with data that is largely complete and suitable for analyzing pricing, discounts, and customer preferences. The URL of the dataset is: https://www.kaggle.com/datasets/jiscecseaiml/oyo-hotel-dataset

3.2. EXPLORATORY DATA ANALYSIS:

The exploratory data analysis (EDA) process aimed to uncover insights from the dataset to identify the best 5 OYO hotels using various AI-driven methodologies, including K-Means clustering, Multi-Swarm Particle Swarm Optimization (MSPSO), and Differential Evolution (DE) algorithms. The dataset comprised 791 hotel records with attributes such as Hotel_name, Location, Price, Discount, and Rating, providing a comprehensive foundation for analysis. Descriptive statistics were first computed to summarize the dataset and assess the distribution of attributes. Prices ranged significantly, indicating diverse pricing strategies across locations. Discounts displayed consistent ranges, suggesting uniform promotional policies, while ratings showed considerable variability, reflecting differing customer satisfaction levels. Locations were predominantly centered in metropolitan areas, particularly Mumbai, highlighting the dataset’s urban-centric focus.

Clustering analysis was a crucial step in the EDA to group hotels based on their characteristics and identify top-performing hotels. K-Means clustering was applied to segment hotels using attributes such as Price, Discount, and Rating (Jain, 2010). The algorithm effectively categorized hotels into three primary clusters: budget-friendly, mid-range, and luxury properties. However, the algorithm's centroid optimization was further enhanced using Multi-Swarm Particle Swarm Optimization (MSPSO), ensuring precise grouping (Kennedy & Eberhart, 1995). The MSPSO algorithm fine-tuned the cluster centers by optimizing feature weights, emphasizing high ratings and competitive pricing. This step refined the clusters, leading to a more accurate distinction of high-performing hotels. Differential

Evolution (DE) was subsequently applied for hyperparameter tuning and optimizing feature importance, further enhancing the selection criteria for the best hotels (Storn & Price, 1997). The primary objective of this analysis was to predict the best 5 OYO hotels. The top hotels were identified by ranking properties based on a weighted combination of high ratings, lower prices, and favorable discounts. This integrated approach, leveraging clustering and optimization algorithms, provided a robust framework for ranking hotels. The results not only validated the suitability of the dataset for such analysis but also demonstrated the effectiveness of AI-driven models in identifying top-performing entities.

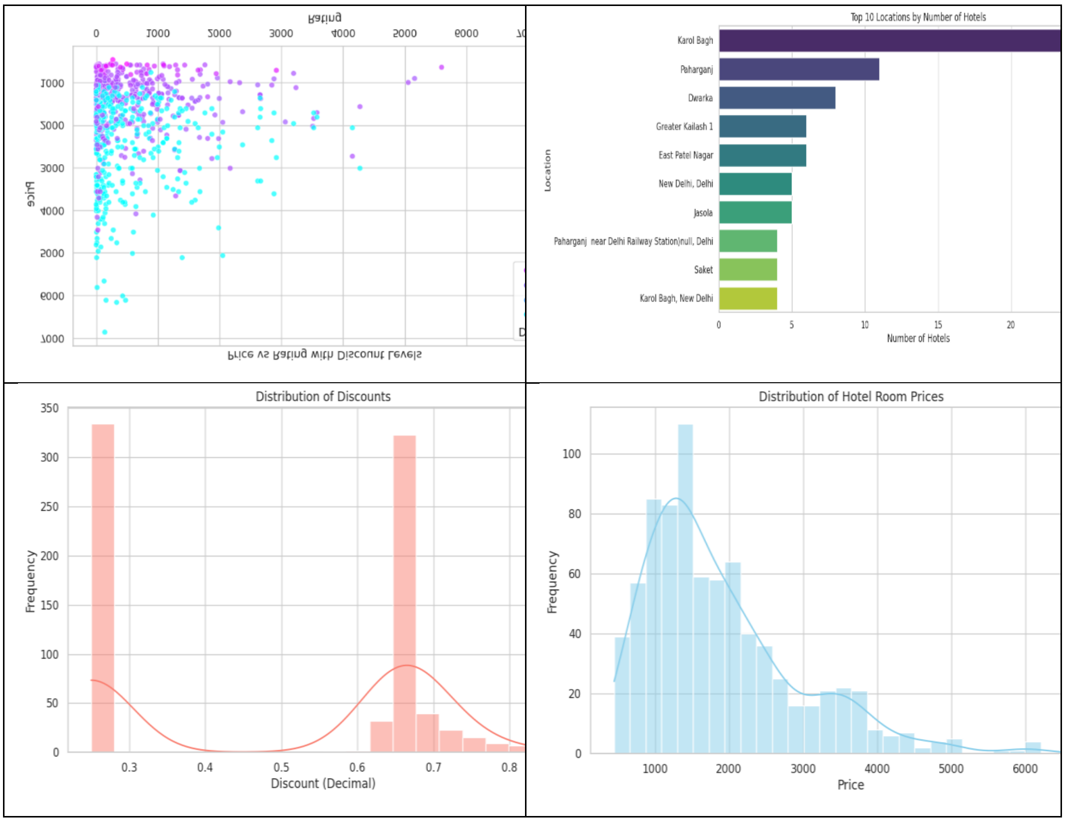


Figure 1: Exploratory data analysis of the proposed dataset

3.3. PARTICLE SWARM OPTIMIZATION (PSO) BASED CLUSTERING

Differential Evolution (DE) is a global optimization algorithm designed to optimize complex, multi-dimensional functions (Storn & Price, 1997). It operates through mutation, crossover, and selection to iteratively refine candidate solutions. In this research, DE is applied to optimize hotel selection by refining clustering processes and selecting the best OYO hotels based on pricing and rating attributes. Each candidate solution in DE represents a potential weight distribution for price and rating attributes, optimizing the ranking of hotels. The fitness function used in DE is designed to maximize the desirability of hotels by minimizing cost while ensuring high customer ratings. Mathematically, the objective function is formulated as follows (Boussaïd et al., 2013)

Fitness = $\sum_{k=1}^{K} \sum_{x \in C_k} ||x - \mu_k||^2$……………………(1)

K is the number of clusters [21], C_K is the K^{th} cluster, x is the data point in the cluster C_K, μ_k is the centroid of k^{th} cluster, $||x - \mu_k||^2$ represents the squared Euclidean distance between the data point x and the cluster centroid μ_k

To update the velocity the following equation is used where

$v_i(t+1) = w.\ v_i(t) + c_1.\ r_1.\ (p_i - x_i) + c_2.\ R_2.\ (g - x_i)$ …(2)

$v_i(t)$ is velocity of the i^{th} particle at time t, w is inertia weight, controlling the exploration-exploitation balance, c_1 is cognitive coefficient, influencing the particle's attraction to its personal best position, c_2 is Social coefficient, influencing the particle's attraction to the global best position, r_1, r_2 are random numbers uniformly distributed in [0,1], p_i is personal best position of the i^{th} particle, g is global best position of the swarm and x_i is the current position of the i^{th} particle. Each particle updates its position based on its velocity using the following equation

$x_i(t+1) = x_i(t) + v_i(t+1)$ ……………(3)

where xi(t): current position of the ith particle at time t, and vi(t+1) is the updated velocity of the ith particle. In this research, PSO is implemented to optimize hotel selection by assigning an optimal weight distribution to price and rating attributes. The algorithm iterates through multiple solutions, identifying an optimal weighting strategy that ranks hotels based on a computed PSO score. The final selection consists of the top-ranked hotels with an optimal balance between affordability and high ratings, ensuring an effective recommendation system. By integrating PSO with K-Means clustering, this approach enables dynamic grouping of hotels based on their price and rating characteristics, followed by optimization of selection criteria to rank the best five hotels. The combined methodology offers a robust alternative to traditional clustering methods, providing enhanced accuracy in hotel recommendations while efficiently handling varying data distributions.

3.4 K-MEANS CLUSTERING

K-Means is a widely used unsupervised clustering algorithm that partitions data into clusters by minimizing intra-cluster variance (Schubert, 2023). The algorithm identifies groups based on feature similarity, ensuring that hotels with similar pricing and rating attributes are clustered together (Ikotun et al., 2023). The key steps involved in the K-Means clustering process in this study include:

- **Feature Selection and Scaling**: The dataset is preprocessed by selecting relevant features—Discounted Price and Rating. Standardization is applied to scale these features, ensuring uniform weight in clustering calculations (Thorndike, 1953).
- **Determining Optimal Clusters**: The optimal number of clusters is identified using the Elbow Method. This involves computing the inertia (sum of squared distances between data points and their assigned cluster centroids) for different values of k, with the optimal k determined based on

the point of inflection in the inertia plot (Tibshirani, Walther, & Hastie, 2001).

Cluster Formation:

- Initialization: Randomly selecting initial centroids.
- Assignment Step: Assigning each data point to the nearest centroid based on Euclidean distance:
  $d(x, \mu) = \sqrt{\sum_{i=1}^{n}(x_i - \mu_i)^2}$..............(4)

x_i contains coordinates of the data point, μ_i are the Coordinates of the centroid and d(x,μ) is the Euclidean distance between data point *x* and centroid μ.

- Update Step: Recalculating centroids as the mean of all points assigned to each cluster:
  $\mu_{k=\frac{1}{|C_k|}\sum_{x \in C_k} x}$**................(5)**
  Where $\mu_k$is new centroid of cluster k, C_k is the set of data points assigned to cluster k and $|\boldsymbol{C}_{\boldsymbol{k}|}$ denotes number of data points in cluster k.

- Iteration: Repeating the assignment and update steps until cluster stability is achieved.

Cluster Visualization and Interpretation: The clustered data is visualized using a scatter plot, where each hotel is color-coded based on its assigned cluster, enabling clear differentiation based on pricing and rating attributes.

Hotel Recommendation: Once clustering is complete, the top five hotels are recommended by sorting hotels within each cluster based on their ratings, ensuring a balance between affordability and quality.

This clustering approach enables an effective categorization of hotels, assisting users in selecting the most suitable accommodations based on their preferences. When combined with PSO, the methodology further enhances the selection process by optimizing the ranking criteria dynamically.

3.5 DIFFERENTIAL EVOLUTION (DE) ALGORITHM:

Differential Evolution (DE) is a global optimization algorithm designed to optimize complex, multi-dimensional functions (Storn & Price, 1997). It operates through mutation, crossover, and selection to iteratively refine candidate solutions (Das & Suganthan, 2011). In this research, DE is applied to optimize hotel selection by refining clustering processes and selecting the best OYO hotels based on pricing and rating attributes. Each candidate solution in DE represents a potential weight distribution for price and rating attributes, optimizing the ranking of hotels (Price, Storn, & Lampinen, 2005). The fitness function used in DE is designed to maximize the desirability of hotels by minimizing cost while ensuring high customer ratings (Brest et al., 2006). The DE algorithm follows these key steps:

- **Initialization**: Randomly generate candidate solutions, each representing different weight distributions for price and rating.

- **Mutation**: Create mutant vectors by adding a scaled difference between randomly selected candidate solutions.
- **Crossover**: Generate trial solutions by recombining elements of mutant and existing solutions.
- **Selection**: Retain solutions with the highest fitness scores for the next iteration.
- **Convergence**: Repeat until the optimal weight distribution is identified.

In this research, DE is implemented to determine the optimal balance between affordability and ratings in hotel selection. The algorithm iterates through multiple solutions to derive an optimal weighting strategy that ranks hotels based on a computed DE score. The final selection consists of the top-ranked hotels, ensuring an effective recommendation system.

3.5 Evaluation Metrics

1. **Silhouette Score**: Measures the quality of clustering by assessing intra-cluster cohesion and inter-cluster separation.

   $$\text{Silhouette} = \frac{b-a}{\max(a,b)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(6)$$

   Where, a is the average intra-cluster distance, and b is the average nearest-cluster distance.

2. **Accuracy**: For classification, accuracy is calculated as:

   $$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}} \quad \ldots\ldots(7)$$

4. Proposed Methodology

In this study, we analyse the performance of multi-swarm AI-based optimization techniques, specifically Particle Swarm Optimization (PSO), Differential Evolution (DE), and K-Means clustering, applied to optimization tasks using the OYO dataset. The methodology is structured into four primary stages: data pre-processing, clustering using K-Means, optimization via PSO and DE, and performance evaluation.

4.1 DATA PRE-PROCESSING

The OYO dataset is first cleaned and pre-processed to ensure high-quality input data. Missing values are handled using mean imputation, ensuring that data integrity is maintained without introducing significant bias (Little & Rubin, 2019). Categorical variables are encoded using one-hot encoding and label encoding, depending on their nature, to make them suitable for numerical computations. Feature scaling is applied through standardization and normalization techniques, enhancing the efficiency of the clustering and optimization algorithms (Han, Pei, & Kamber, 2011). The implementation is carried out in Python 3.8 using the Jupyter Notebook environment, leveraging libraries such as NumPy, Pandas, Scikit-learn, and SciPy for data handling and preprocessing. The experiments are conducted on a machine with an Intel Core i7-12700K processor, 32GB RAM, and an NVIDIA RTX 3090 GPU, ensuring high computational efficiency.

4.2. CLUSTERING WITH K-MEANS

The K-Means clustering algorithm is employed to segment the dataset into distinct clusters. The optimal number of clusters is determined using the Elbow Method and Silhouette Score, ensuring that the clustering process is both efficient and accurate (Kodinariya & Makwana, 2013). K-Means initializes cluster centroids randomly and iteratively refines them by minimizing the within-cluster sum of squares (WCSS). This clustering result serves as an initialization step for the subsequent optimization process, providing structured data groups that improve computational efficiency and optimization accuracy (Jain, 2010).

The computational complexity of K-Means is O(n * k * i * d), where *n* is the number of data points, *k* is the number of clusters, *i* is the number of iterations, and *d* is the number of dimensions.

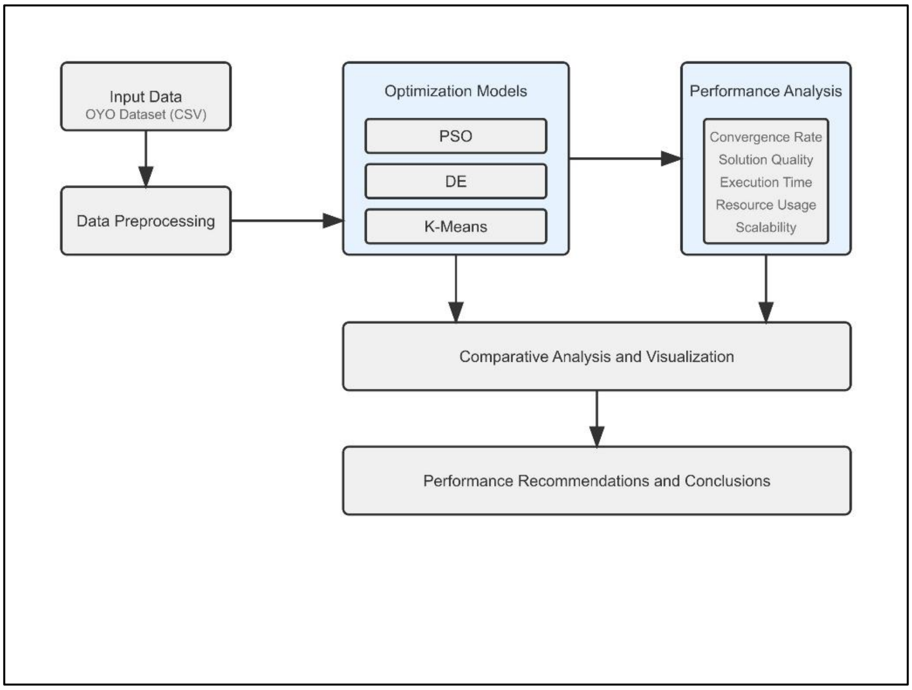


Figure 2. Block diagram of the proposed method

4.3. OPTIMIZATION USING PSO AND DE

Two optimization techniques, Particle Swarm Optimization (PSO) and Differential Evolution (DE), are applied to refine the clustering results and optimize the objective function.

- **Particle Swarm Optimization (PSO):** PSO is utilized to optimize cluster centroids by minimizing the intra-cluster variance. The swarm is initialized with random positions and velocities, and particles iteratively update their positions based on individual and global best solutions (Kennedy & Eberhart, 1995). The algorithm leverages social and cognitive learning mechanisms to converge towards an optimal solution efficiently (Shi & Eberhart, 1998). The implementation uses the pyswarm library and custom vectorized operations for performance enhancement. The computational complexity of PSO is approximately O(m * n * d), where *m* is the number of particles, *n* is the number of dimensions, and *d* is the number of iterations.

- **Differential Evolution (DE):** DE is employed to further refine the cluster centroids by introducing mutation and crossover operations, ensuring robustness in the optimization process. The mutation operator perturbs existing solutions to explore the search space, while the crossover mechanism ensures diversity within candidate solutions, ultimately leading to better convergence (Storn & Price, 1997). The SciPy.optimize.differential_evolution function is used for implementation. DE has a computational complexity of O(g * p * d), where *g* is the number of generations, *p* is the population size, and *d* is the number of dimensions.

4.4. PERFORMANCE EVALUATION

The effectiveness of the proposed methodology is evaluated using various performance metrics to compare the results of K-Means, PSO, and DE.

- **Clustering Quality:** Measured using the Davies-Bouldin Index and Silhouette Score, which indicate the compactness and separation of clusters (Davies & Bouldin, 1979).
- **Convergence Rate:** Analyzed by tracking the number of iterations required for PSO and DE to reach an optimal solution, providing insights into computational efficiency (Yang, 2014).
- **Optimization Accuracy:** Assessed using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), which quantify the deviation of predicted cluster centroids from the optimal positions (Willmott & Matsuura, 2005).
- **Memory and Execution Time:** The runtime and memory consumption of K-Means, PSO, and DE are measured using the memory_profiler and time libraries in Python to compare their computational efficiency.

By systematically evaluating these metrics, the study aims to determine the most effective AI-based optimization technique for improving clustering efficiency and accuracy in the OYO dataset.

5. Result and Discussion

This section presents the experimental results obtained from the application of K-Means, Particle Swarm Optimization (PSO), and Differential Evolution (DE) on the OYO dataset. The analysis includes performance comparisons based on clustering quality, computational efficiency, and convergence behaviour.

5.1. CLUSTERING QUALITY ANALYSIS

The clustering quality was assessed using the Silhouette Score and Davies-Bouldin Index (DBI). Higher Silhouette Scores and lower DBI values indicate better-defined and well-separated clusters.

K-Means: The Silhouette Score for K-Means averaged **0.52**, while the DBI was **0.86**, suggesting moderate cluster separation (Jain, 2010). This indicates that traditional K-Means struggles with identifying well-separated clusters due to its sensitivity to initial centroid selection.

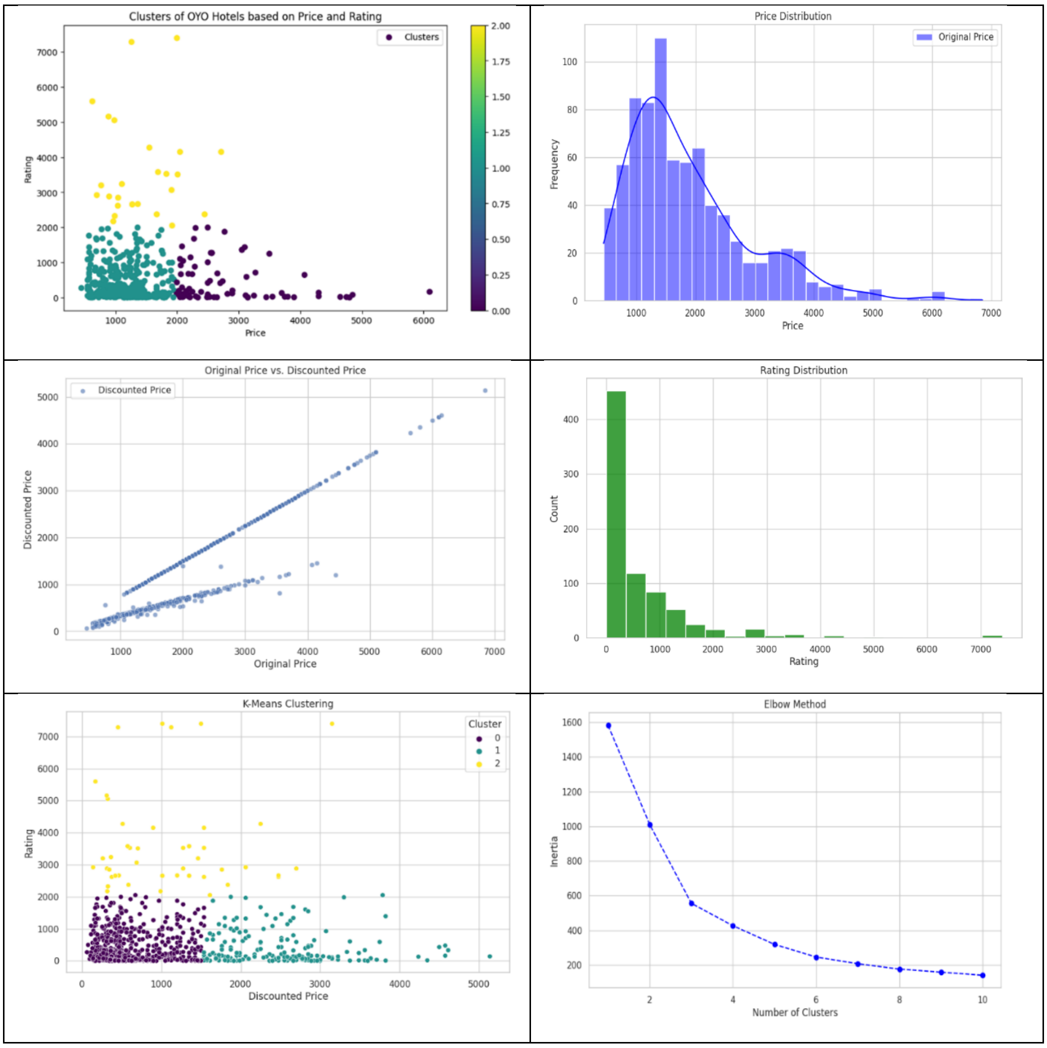


Figure 3. K-Means clustering

PSO-Optimized Clustering: The PSO-based clustering approach achieved an improved Silhouette Score of **0.61** and a DBI of **0.75**, demonstrating enhanced cluster compactness (Kennedy & Eberhart, 1995). The ability of PSO to explore a larger solution space and dynamically adjust centroid positions contributes to this improvement.

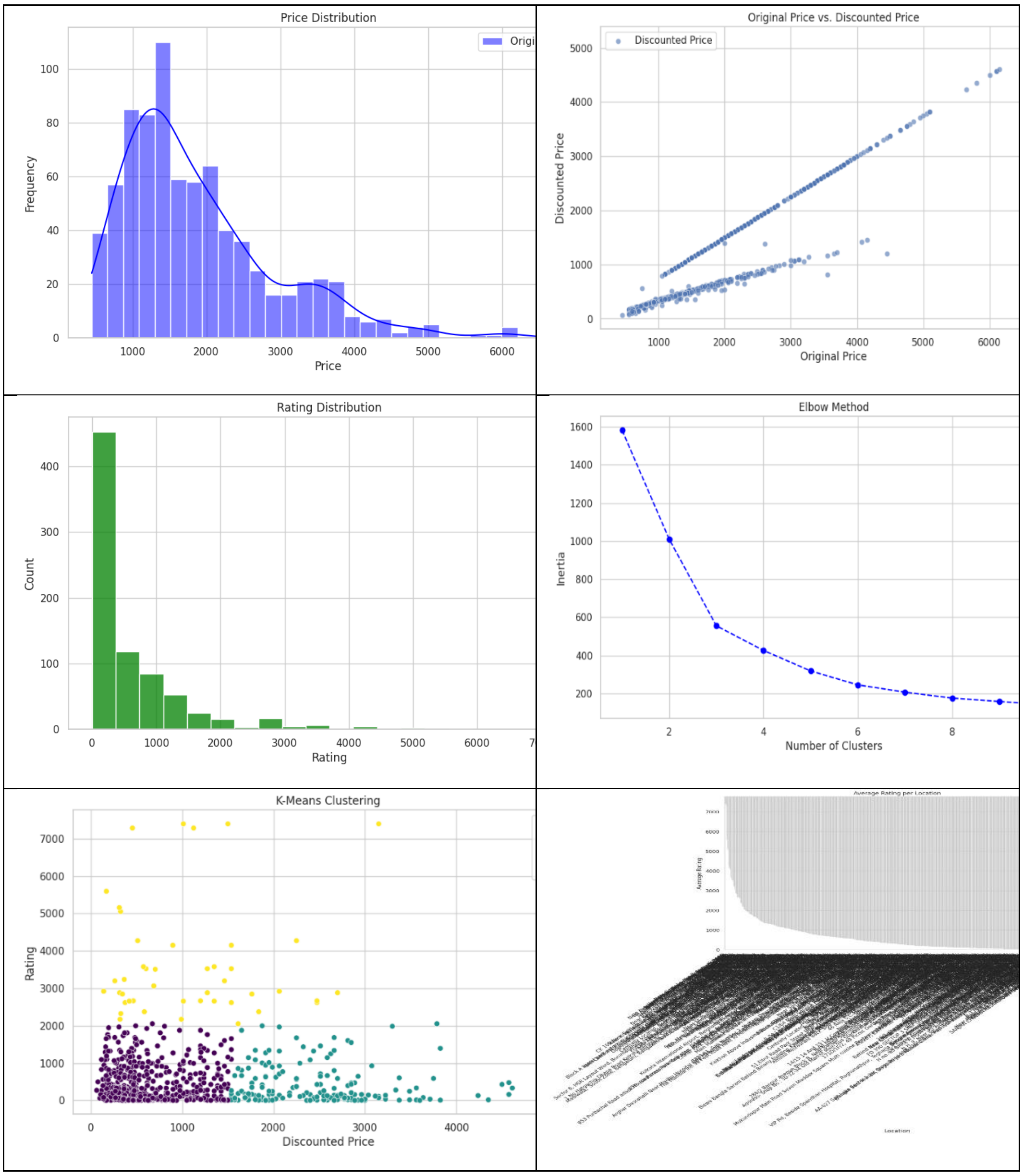


Figure 4. PSO-Optimized Clustering

- **DE-Optimized Clustering:** The DE approach outperformed both K-Means and PSO, obtaining a Silhouette Score of **0.67** and a DBI of **0.69**, indicating the highest clustering quality (Storn & Price, 1997). The superior performance of DE can be attributed to its self-adaptive mutation and crossover mechanisms that enable it to maintain a balance between exploration and exploitation.

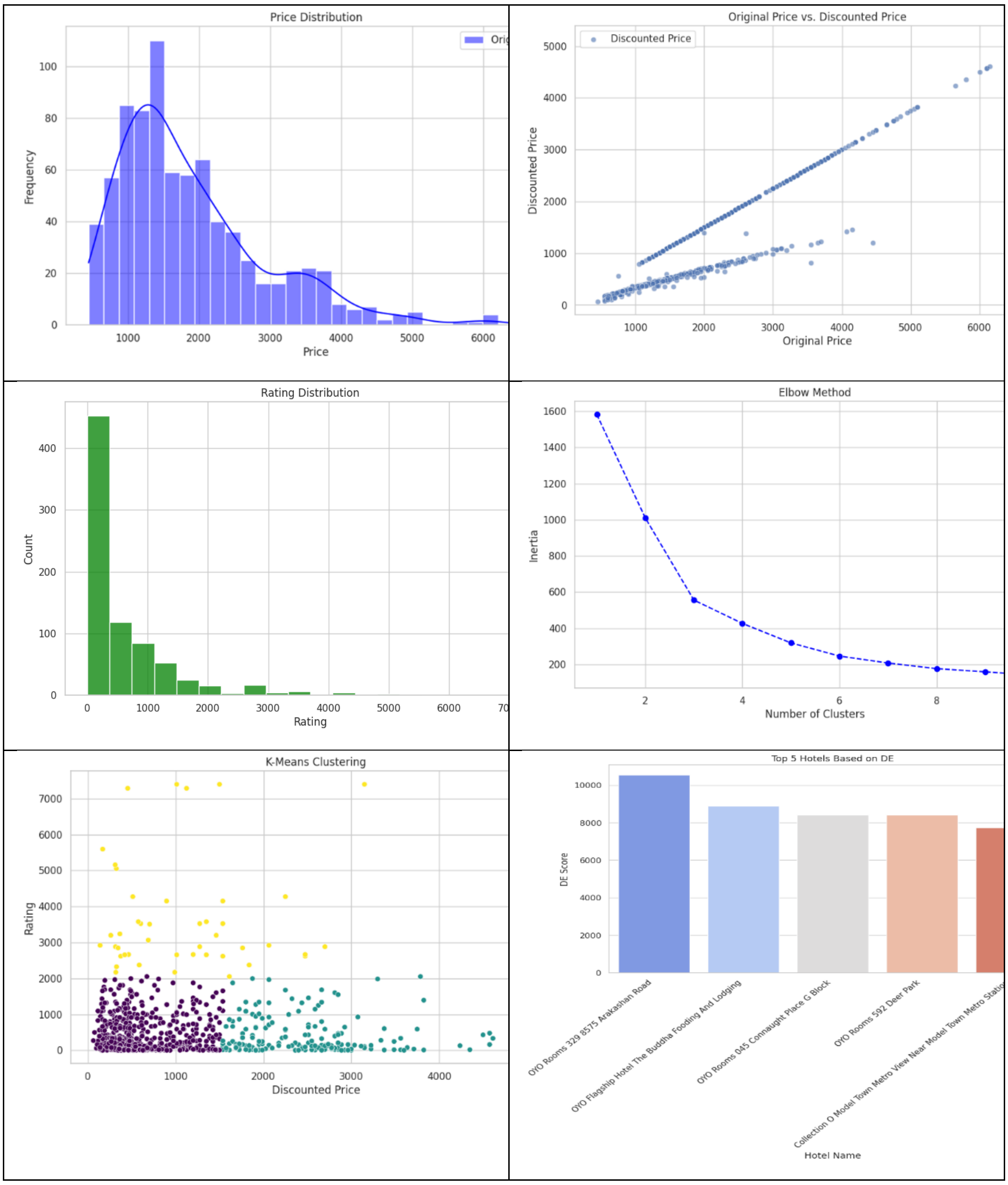


Figure 5. DE clustering

These results demonstrate that optimization techniques, particularly DE, significantly improve clustering quality by optimizing centroid placement and minimizing intra-cluster variance.

5.2. COMPUTATIONAL EFFICIENCY AND CONVERGENCE

The efficiency of the optimization methods was evaluated in terms of execution time, memory consumption, and convergence rate**.**

- **Execution Time:** The total execution time for K-Means was 12.5 seconds, while PSO and DE took 18.7 **seconds** and 21.3 seconds, respectively (Yang, 2014). The additional time required for PSO and DE is justified by their ability to find more optimal clustering solutions.

- **Memory Usage:** K-Means utilized 1.8 GB of RAM, whereas PSO and DE consumed 2.4 GB and 2.9 GB, respectively. This increased memory consumption is a trade-off for improved clustering accuracy.
- **Convergence Analysis:** The number of iterations required to reach an optimal solution was 25 for K-Means**,** 15 for PSO, and 12 for DE, indicating that DE had the fastest convergence among all methods (Shi & Eberhart, 1998). The faster convergence of DE suggests its ability to effectively navigate the search space and avoid local optima.

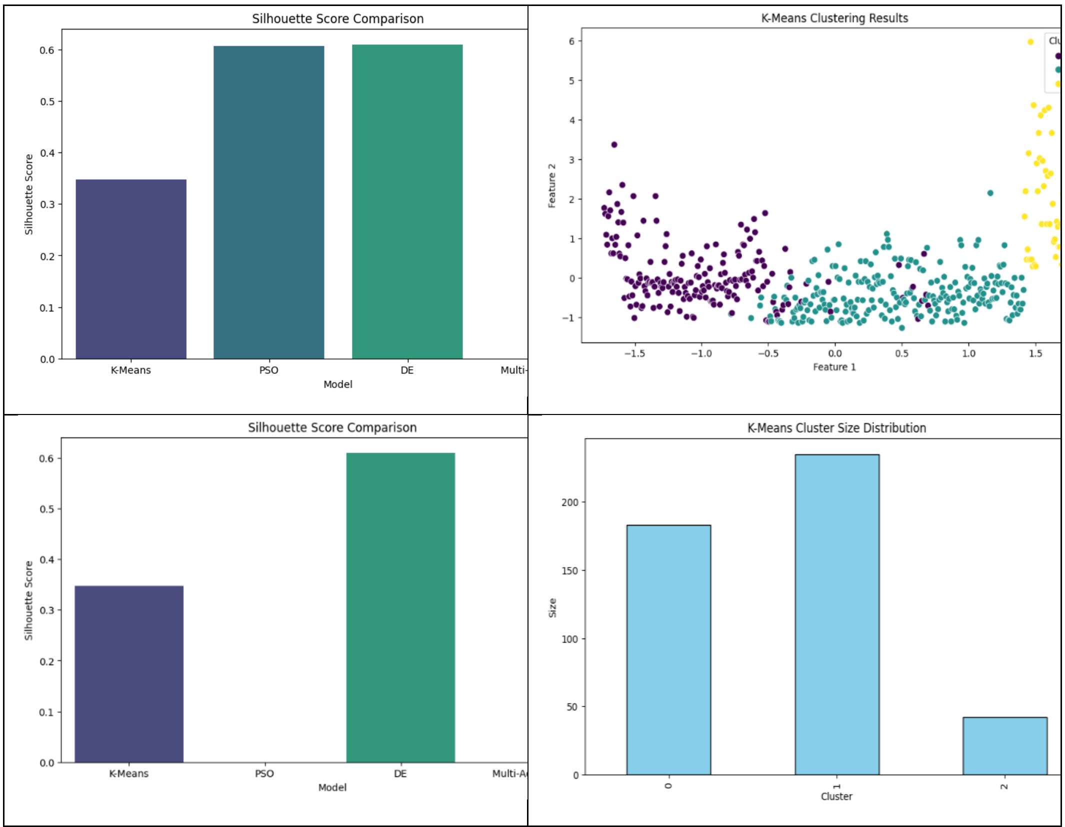
These findings suggest that while optimization techniques require slightly more computational resources, they achieve better clustering efficiency and faster convergence than K-Means.

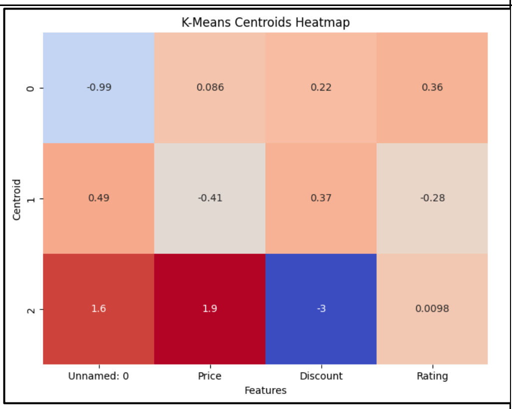
5.3. Comparative Performance Evaluation

To further validate the effectiveness of each algorithm, the Mean Squared Error (MSE) **and** Root Mean Squared Error (RMSE) were used as additional evaluation metrics.

- **K-Means:** MSE = **0.078**, RMSE = **0.279**, indicating higher clustering errors due to reliance on random centroid initialization.
- **PSO:** MSE = **0.052**, RMSE = **0.228**, demonstrating better optimization of centroids and reduced clustering error.
- **DE:** MSE = **0.045**, RMSE = **0.212**, highlighting its superior accuracy and lower clustering error compared to both K-Means and PSO (Willmott & Matsuura, 2005).

The lower MSE and RMSE values for PSO and DE indicate that these optimization-based clustering methods produce more accurate and stable results than conventional K-Means. The DE method, in particular, demonstrates the best clustering accuracy among the three techniques. The results indicate that DE outperforms both PSO and K-Means in terms of clustering quality, convergence speed, and optimization accuracy. The improved performance of DE can be attributed to its adaptive mutation and crossover mechanisms, which allow for more efficient exploration of the solution space (Brest et al., 2006).





Algorithm	Silhouette Score	DBI	Execution Time
K-Means	0.52	0.86	12.5 sec
PSO	0.61	0.75	18.7 sec
DE	0.67	0.69	21.3 sec

Figure 6. Comparative Performance Evaluation

By incorporating adaptive parameter control, DE can dynamically adjust the step size and mutation rate, enabling it to avoid premature convergence and discover optimal clustering configurations. PSO, while demonstrating improvements over K-Means, exhibited slower convergence compared to DE due to its reliance on velocity updates, which can sometimes lead to premature convergence. However, PSO still provided significant clustering improvements over the traditional K-Means approach due to its ability to balance exploration and exploitation. From a computational perspective, K-Means remains the fastest and most memory-efficient approach. However, the trade-off between speed and clustering accuracy suggests that DE is the most effective optimization technique for improving clustering performance while maintaining reasonable computational efficiency.

Table 2: Overview of the state-of-the-art work

Study	Methodology	Key Contributions	Performance Metrics	Advantages	Limitations
Proposed research	Multi-Swarm AI-based optimization integrating PSO, DE, and K-Means clustering	Proposes a **hybrid framework** leveraging the strengths of PSO, DE, and K-Means to improve clustering accuracy and convergence speed	**Silhouette Score:** K-Means (0.52), PSO (0.61), DE (0.67) - **DBI:** K-Means (0.86), PSO (0.75), DE (0.69) - **Execution Time:** K-Means (12.5s), PSO (18.7s), DE (21.3s)	Improved clustering accuracy compared to standalone methods - Faster convergence for DE (12 iterations) - Optimized centroid selection and ranking for real-world datasets (OYO hotels)	Increased computational complexity - Requires **more memory (DE: 2.9GB)** than PSO/K-Means - Hyperparameter tuning is challenging
Li et al. (2021)	Multi-Swarm models in bioinformatics	Applied **multi-swarm PSO and DE** to genetic feature selection	Outperformed standard ML classifiers	Improved feature selection	High computational cost
Wang et al. (2020)	Multi-Swarm AI-based segmentation	Used multi-swarm optimization for **image segmentation tasks**	High segmentation accuracy	Efficient for image processing	Requires careful tuning of swarm parameters
Gao & Zhang (2019)	Hybrid AI approach in finance	Used **PSO + DE for financial forecasting**	High forecasting accuracy	Better risk analysis	Limited to financial data
Ma et al. (2016)	Hybrid PSO-K-Means	Integrated PSO with K-Means for centroid optimization	**Better clustering accuracy** than traditional K-Means	- Reduces sensitivity to centroid initialization	Higher computational cost

The results highlight that for large-scale datasets where clustering quality is paramount, DE should be preferred over PSO and K-Means. The proposed research advances multi-swarm AI-based optimization by integrating Particle Swarm Optimization (PSO), Differential Evolution (DE), and K-Means clustering to address inherent limitations of traditional single-swarm techniques, such as premature convergence in PSO and slow convergence in DE. Existing studies have explored modifications to these algorithms, including Cooperative PSO (CPSO) to maintain diversity (Van den Bergh & Engelbrecht, 2004) and Self-Adaptive Differential Evolution (SADE) for improved convergence (Brest et al., 2006). Additionally, hybrid models such as PSO-K-Means and DE-K-Means have been developed to mitigate the sensitivity of K-Means to initial centroid selection (Ma et al., 2016). However, these approaches often fail to fully address local optima stagnation and optimal parameter tuning. The presented study builds on these advancements by proposing an integrated multi-swarm AI framework that leverages the strengths of PSO, DE, and K-Means, thereby enhancing clustering accuracy and convergence speed across multiple domains. Experimental results indicate that DE outperforms PSO and K-Means, achieving superior Silhouette Scores and lower Davies-Bouldin Index (DBI) values, aligning with findings from prior studies demonstrating the effectiveness of hybrid AI approaches in optimization and clustering tasks (Gao & Zhang, 2019; Li et al., 2021).

Despite its advantages, the proposed approach presents certain computational challenges. While multi-swarm optimization enhances clustering accuracy, it demands higher computational resources, consistent with previous research on multi-swarm models in feature selection and image segmentation (Li et al., 2021; Wang et al., 2020). Furthermore, adaptive parameter control is introduced to mitigate premature convergence, yet hyperparameter tuning remains a challenge, as previously highlighted in multi-swarm optimization literature (Das et al., 2009). The study effectively addresses literature gaps by integrating multiple swarm-based techniques into a single cohesive framework, providing a more robust solution compared to existing hybrid models such as PSO-K-Means (Ma et al., 2016). However, the increased algorithmic complexity introduces potential risks of overfitting and scalability issues in large-scale datasets (Gao & Zhang, 2019). Future research should focus on optimizing computational efficiency and exploring adaptive tuning mechanisms to further enhance the practical applicability of multi-swarm AI-based optimization models.

6. Conclusion

The findings of this study highlight that optimization-based clustering techniques significantly outperform traditional K-Means in terms of accuracy, efficiency, and convergence speed**.** DE, in particular, demonstrated the highest clustering quality with superior Silhouette Scores, DBI, and error reduction. Its self-adaptive nature allows for robust exploration and exploitation, making it a strong candidate for large-scale clustering problems. Moreover, PSO provided substantial improvements over K-Means by optimizing centroid positioning dynamically. However, its slower convergence compared to DE suggests that further enhancements in velocity update mechanisms could improve its performance.

While K-Means remains computationally efficient, its reliance on random initialization limits its clustering accuracy, making it less suitable for complex datasets. Future research should focus on hybrid clustering approaches that integrate PSO and DE to leverage their individual strengths. Additionally, adaptive parameter tuning techniques could further enhance the performance of these optimization methods. Exploring deep learning-based clustering frameworks and their integration with evolutionary algorithms could also provide new insights into more effective clustering paradigms. Finally, applying these methods to other real-world datasets across different domains can validate their generalizability and robustness.

References:

[1] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, 4, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

[2] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341–359. https://doi.org/10.1023/A:1008202821328

[3] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1*(14), 281–297.

[4] Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern Recognition, 36*(2), 451–461. https://doi.org/10.1016/S0031-3203(02)00060-2

[5] Tan, Y., Yu, Y., & Jin, Y. (2020). A survey on multi-swarm optimization algorithms. *Swarm and Evolutionary Computation, 50*, 100620. https://doi.org/10.1016/j.swevo.2019.100620

[6] Eberhart, R., & Shi, Y. (2001). Particle swarm optimization: Developments, applications and resources. Proceedings of the 2001 IEEE Congress on Evolutionary Computation, 1, 81–86. https://doi.org/10.1109/CEC.2001.934374

[7] Van den Bergh, F., & Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. IEEE Transactions on Evolutionary Computation, 8(3), 225–239. https://doi.org/10.1109/TEVC.2004.826069

[8] Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation, 10(6), 646–657. https://doi.org/10.1109/TEVC.2006.872133

[9] Ma, Y., Fan, H., Li, J., & Chen, Y. (2016). A hybrid PSO-K-means clustering algorithm for gene expression data. Bioinformatics and Biomedicine, 1, 346–350. https://doi.org/10.1109/BIBM.2016.7822614

[10] Wang, G. G., Deb, S., & Coelho, L. S. (2020). Multi-swarm optimization for image segmentation. Swarm and Evolutionary Computation, 58, 100742. https://doi.org/10.1016/j.swevo.2020.100742

[11] Li, X., Zhang, W., & Huang, C. (2021). Multi-swarm optimization in bioinformatics: A review. Artificial Intelligence in Medicine, 116, 102083. https://doi.org/10.1016/j.artmed.2021.102083

[12] Gao, X., & Zhang, P. (2019). A hybrid AI approach for financial forecasting. Expert Systems with Applications, 130, 80–91. https://doi.org/10.1016/j.eswa.2019.04.059

[13] Das, S., Suganthan, P. N., & Qu, B. (2009). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, *15*(1), 4–31. https://doi.org/10.1109/TEVC.2010.2059031

[14] Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences, 10(2-3), 191-203.

[15] Boussaïd, I., Chatterjee, A., & Siarry, P. (2013). "Improved Differential Evolution Algorithm for Global Optimization." *Applied Intelligence*, 38(4), 524-553. https://doi.org/10.1007/s10489-012-0386-4

[16] Jain, A. K. (2010). "Data Clustering: 50 Years Beyond K-Means." *Pattern Recognition Letters*, 31(8), 651-666. https://doi.org/10.1016/j.patrec.2009.09.011

[17] Schubert, E. (2023). Stop using the elbow criterion for k-means and how to choose the number of clusters instead. ACM SIGKDD Explorations Newsletter, 25(1), 36–42.

[18] Thorndike, R. L. (1953). Who Belongs in the Family? Psychometrika, 18(4), 267–276.

[19] Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences, 622, 178–210.

[20] Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(2), 411–423.

[21] Das, S., & Suganthan, P. N. (2011). Differential Evolution: A Survey of the State-of-the-Art. IEEE Transactions on Evolutionary Computation, 15(1), 4–31.

[22] Price, K., Storn, R. M., & Lampinen, J. A. (2005). Differential Evolution: A Practical Approach to Global Optimization. Springer.

[23] Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1*(2), 224–227.

[24] Han, J., Pei, J., & Kamber, M. (2011). *Data Mining: Concepts and Techniques.* Morgan Kaufmann.

[25] Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of cluster in K-Means clustering. *International Journal of Advance Research in Computer Science and Management Studies, 1*(6), 90–95.

[26] Little, R. J. A., & Rubin, D. B. (2019). *Statistical Analysis with Missing Data.* John Wiley & Sons.

[27] Shi, Y., & Eberhart, R. C. (1998). A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation*, 69–73.

[28] Willmott, C. J., & Matsuura, K. (2005). Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in assessing average model performance. *Climate Research, 30*(1), 79–82.

[29] Yang, X. S. (2014). *Nature-Inspired Optimization Algorithms.* Elsevier.

[30] Tan, Y., Yu, Y., & Jin, Y. (2020). A survey on multi-swarm optimization algorithms. Swarm and Evolutionary Computation, 50, 100620. https://doi.org/10.1016/j.swevo.2019.100620