INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Check for updates

OPEN ACCESS

Received: 26-10-2024

Accepted: 23-11-2024

Published: 06-12-2024

Citation: Bangoria BM, Panchal SS, Panchal SR, Bangoria TM (2024) A Novel Machine Learning Approach for Multidimensional Dynamic Destination Recommender S earch System Employing Clustering using Optimization Techniques. Indian Journal of Science and Technology 17(44): 4679-4693. https://doi.org/10.17485/IJST/v17i44.3525

* **Corresponding author**.

bhoomi.bangoria@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

A Novel Machine Learning Approach for Multidimensional Dynamic Destination Recommender S earch System Employing Clustering using Optimization Techniques

Bhoomi Mansukhlal Bangoria[1]*, Sweta S Panchal[2], Sandipkumar R Panchal[3], Tusharkumar Mansukhbhai Bangoria[4]

1 Research Scholar, Computer Engineering, Dr. Subhash University, Junagadh, Gujarat, India
2 Associate Professor, EC Department, Dr. Subhash University, Junagadh, Gujarat, India
3 Professor, EC Department, Dr. Subhash University, Junagadh, Gujarat, India
4 Associate Vice President, Product Engineering Services, Moschip Technologies Limited, Ahmedabad, Gujarat, India

Abstract

Objectives: To find the most suitable destinations according to a selection of various dimensions by the user. Optimization techniques are applied to clustering with the help of various objective functions, to find optimal clusters for better recommendations. **Methods**: This approach uses a hybrid filtering system for recommendation with a weighted K-means clustering algorithm and uses optimization techniques to improve cluster formation recommendation accuracy. This study used a data set from Kaggle, which considers different city names, types, and significance. According to city names, using a geolocator object in Python gets latitude and longitude for precise clustering. **Findings**: The elbow method utilizes the K-means and weighted K-means clustering algorithm to determine the number of clusters. We use optimization techniques such as the Artificial Bee Colony (ABC) algorithm and Particle Swarm Optimisation (PSO) to improve the cluster formation recommendation accuracy, as demonstrated by the following results: The accuracy of K-means with ABC and PSO is respectively 85% and 88%, while weighted K-means with ABC and PSO is respectively 90% and 93%. **Novelty**: This study highlights how advanced optimization techniques like ABC and PSO enhance K-means and weighted K-means clustering accuracy, precision, and recall. The combination of weighted K-means and PSO further enhances performance, making it the perfect choice for tasks that demand high-quality clustering and recommendation systems. Compared to the K-means clustering algorithm with PSO optimization, the ratio of successfully recommended relevant destinations is 7% higher.

Keywords: Recommender System; Clustering; Destination Recommender System (DRS); Machine Learning; Weighted Kmeans clustering Algorithm; ABC; PSO

1 Introduction

Most people travel between various geographical locations at some point in their lives. Some people travel for work-related or connected reasons, while others do so for relaxation, discovery, or enjoyment. Furthermore, some people may move via traveling, perhaps to a different country or city. Whatever the reason for their trip, a lot of people want housing choices, such as hotels or restaurants. While touring a new area or attending meetings and events, guests can stay in comfortable and useful short-term accommodations of this type. In the end, travel can be a beneficial experience that aids in horizon expansion and the discovery of new locations and cultures. Technology and Recommendation Systems (RS) for travel can help people. Tourists increase the economy of organized communities; they also provide local along with global employment. Utilization of DRS to increase visitor satisfaction has become much more common in recent years.

Strong new technologies, known as recommender systems, let consumers find the most popular tourist spots according to their interests[1]. We aim to create tourism recommendation systems by utilizing multi-objective optimization algorithm techniques and exploring various multi-criteria methods. This approach will facilitate the creation of more customized systems that are more responsive to users' interests and preferences[2]. The study's results show that the improved K-means clustering algorithm makes the e-commerce recommendation system much better at making suggestions and more accurate than other similar algorithms[3]. Using the k-means algorithm and GA, an innovative approach optimizes and classifies tourism objectives to recommend the optimal tourism path. There is still room for improvement in using ML algorithms[4]. The harmony search (HS) and ABC metaheuristic optimization algorithms both with K-means have been first developed in this study and can be used to choose the best cluster centroids to highpoint favorable targets in the prospecting stage of mineral explorations[5]. The study provides the best recommendations based on ratings; however, for further experiments, it is necessary to incorporate all attribute elements from tourism data[6].

AI has significantly enhanced the capabilities of recommender systems, enabling them to process large datasets for personalized recommendations[7]. However, despite advancements, current systems often fail to incorporate multi-dimensional user preferences, a gap that this study addresses through optimization techniques applied to the weighted K-means clustering.

Reviewing various papers reveals the rarity of personalized RS in the tourism industry. The presented research work focuses on selecting various dimensions based on the user's choice, using optimization techniques. Recommender Systems have significantly enhanced tourism experiences by enabling users to find destinations based on personalized preferences. Despite advancements, current systems often fail to incorporate multi-dimensional user preferences comprehensively. This research bridges that gap by utilizing weighted K-means clustering enhanced by Artificial Bee Colony (ABC) and Particle Swarm Optimization (PSO) techniques. This approach highlights the importance of multi-objective optimization in tourism recommendation systems.

The mean of precision and recall, which provides a balanced measure of RS performance, is 6% higher in the weighted K-means with PSO compared to the traditional K-means with PSO optimization.

2 Methodology

This study compares two clustering algorithms: the K-means clustering algorithm and the weighted K-means clustering algorithm[1]. This study integrates optimization techniques (ABC and PSO) with weighted K-means clustering to address multidimensional user preferences in destination recommendation systems. The parameters for ABC include Colony Size (20), Employed Bees (10), and Maximum Iterations (100). For PSO, parameters include Particles (30), Inertia Weight (0.7), and Cognitive and Social Components (2). These configurations were chosen based on their proven effectiveness in prior studies. This study used a data set (https://www.kaggle.com/datasets/saketk511/travel-dataset-guide-to-indias-must-see-places), which considers different city names, types, and significance. Using a geolocator object in Python, we can obtain latitude and longitude based on the city names.

The study shows that future work will focus on the artificial bee colony algorithm and the hybrid attribute clustering algorithm[8]. Fitness selection considers the distance between the sample centers, thereby reducing the likelihood of excessive clustering in the sample distribution[9]. Use the ABC method to improve clustering by comparing proposed cluster centers and the similarity of data points and reassigning clusters accordingly[10]. MapReduce-based optimization methods enhance search quality and local search time by combining the features of both MapReduce and specific methods and employing various optimization techniques to achieve superior results[11]. While prior studies employed the ABC and PSO algorithms independently, this study integrates these with weighted K-means clustering to optimize feature weighting and clustering efficiency. The key novelty lies in adapting the optimization techniques to address multidimensional preferences in recommendation systems.

ABC and PSO were selected for their ability to explore solution spaces effectively while minimizing convergence issues common in high-dimensional clustering problems.

2.1 K-means Clustering Algorithm with ABC Optimization Algorithm

This hybrid approach enables more flexible cluster optimization, focusing instead on minimizing the Euclidean (or other) distances between data points and centroids.

Algorithm:

Step 1: Initialise Parameters

- Input: Data points x = $\{x_1, x_2, x_3, \ldots, x_n\}$, number of clusters K, and ABC parameters.
- Colony size (number of bees) SN
- Number of employed bees NE
- Number of onlooker bees NO
- Maximum number of iterations MaxIter
- Limit refers to the maximum number of trials for a solution before abandoning it.
- Objective function: distance-based fitness function

Step 2: Initialise Food Sources (Solutions)

- Randomly initialize food sources (possible solutions) by selecting K cluster centroids from the dataset.
- $S_i = \{C_{i1}, C_{i2}, \ldots, C_{iK}\}$ for each food source i.
- Each C_{iK} is a d-dimensional centroid, where d is the number of features in the data.

Step 3: Evaluate Fitness of Initial Food Sources

- For each S_i solution, assign each data point to the nearest centroid using the Euclidean distance:

$$d(x_j, C_k) = ||x_j - C_k|| \tag{1}$$

where C_k is the k-th centroid.

- Compute the fitness of each food source by averaging the distances between points and their nearest centroid:

$$F_i = 1/n \sum_{j=1}^{n} min\, d(x_j, C_k) \tag{2}$$

Lower fitness values indicate better solutions (closer distances).
Step 4: Employed Bee Phase

- Each employed bee modifies its solution S_i by updating the position of centroids. A new solution S_i‘ is generated by:

$$C'_{ij} = C_{ij} + \emptyset_{ij}\,(C_{ij} - C_{kj}) \tag{3}$$

where k is a random index different from i, and $\emptyset_{ij}$ is a random number between [−1,1].

- Assign data points to the nearest centroids for the new solution S_i‘ and calculate the new fitness using the distance-based metric.
- If the new solution performs better, it should be substituted for the old solution S_i.

Step 5: Onlooker Bee Phase

- Each onlooker bee chooses a food source S_i based on its possibility, which is proportional to its fitness:

$$P_i = \frac{F_i}{\sum_{j=1}^{SN} F_j} \tag{4}$$

- Modify the selected food source likewise to the employed bee phase. Evaluate the new solution and update it if the fitness improves.

Step 6: Scout Bee Phase

- If a solution does not improve for a certain number of trials (limit), a scout bee abandons it and generates a new random solution to replace the abandoned one.

Step 7: Update the Best Solution

- Track the best solution (centroid positions) found by all the bees based on the lowest distance-based fitness value.

Step 8: Repeat until Stopping Condition

- Continue Steps 4–7 until you reach the MaxIter maximum number of iterations or meet another stopping criterion.

Step 9: Final K-means Refinement

- Perform a final iteration of the K-means algorithm using the best centroids discovered by ABC to refine the cluster assignments.

Step 10: Output

- We have assigned the final cluster centroids and data points.

2.2 K-means Clustering Algorithm with PSO Optimization Algorithm

This hybrid approach combines the investigation capabilities of PSO with the local modification of K-means, providing an efficient clustering method. Instead, the algorithm directly minimizes the addition of the distances between the data points and their nearest centroids.

Algorithm:

Step 1: Initialise Parameters

- Input: Data points x = {x_1, x_2, x_3, … , x_n}, number of clusters K, and PSO parameters.
- Number of Particles P
- Maximum number of iterations MaxIter
- Inertia weight w
- Cognitive component c_1
- Social component c_2
- Random coefficients r_1 and r_2

Step 2: Initialise Particles (Solutions)

- Each particle represents a set of cluster centroids. Randomly initialize the positions of the particles p_i = {C_{i1}, C_{i2}, …, C_{iK}}, where each C_{iK} is a d-dimensional centroid (where d is the number of features in the data).
- Set the initial velocities of the particles v_i to zero.
- Set the personal best of each particle $pbest_i$ to its initial position.
- Set the global best gbest to the position of the particle with the best fitness value.

Step 3: Estimate the Initial Fitness of Particles

- For every particle p_i, assign each data point to its nearest centroid using the Euclidean distance.

$$d(x_j, C_k) = ||x_j - C_k|| \tag{5}$$

- Compute the fitness of individual particle p_i based on the average (or total) distance between data points and their assigned centroids:

$$F_i = 1/n \sum_{j=1}^{n} min\, d(x_j, c_k) \tag{6}$$

Lower fitness values indicate better solutions (closer distances).

Step 4: Update Personal and Global Best

- For each particle p_i:
- If the fitness of p_i is better than the fitness of its personal best $pbest_i$, update $pbest_i$ = p_i
- If the fitness of p_i is better than the fitness of its global best gbest, update gbest = p_i

Step 5: Update Particle Velocities and Positions

- For each particle p_i, modernize its velocity v_i and position p_i using the PSO update equations:

$$v_i\,(t+1) = w * v_i(t) + c_1 * r_1 * (pbesti - pi(t)) + c_2 * r_2 * (gbest - pi(t)) \tag{7}$$

$$p_i(t+1) = p_i(t) + v_i(t+1) \tag{8}$$

where:
$v_i(t)$ is the velocity of particle i at repetition t,
$p_i(t)$ is the centroids of particle i at repetition t,
$pbest_i$ is the particle i, personal best position,
gbest is the all particles' global best position,
w is the inertia weight that controls the impact of the prior velocity,
c_1 and c_2 are cognitive and social constants, and
r_1 and r_2 are random numbers in the range [0, 1]
Step 6: Repeat until Stopping Condition

- Continue Steps 3–5 until you reach the MaxIter maximum number of iterations or meet another stopping criterion, such as a threshold for fitness improvement or convergence.

Step 7: Final K-means Refinement

- Perform a final K-means clustering step using the best centroids gbest found by PSO to refine the cluster assignments.

Step 8: Output

- Final cluster centroids and the data point assignments.

2.3 Proposed Work

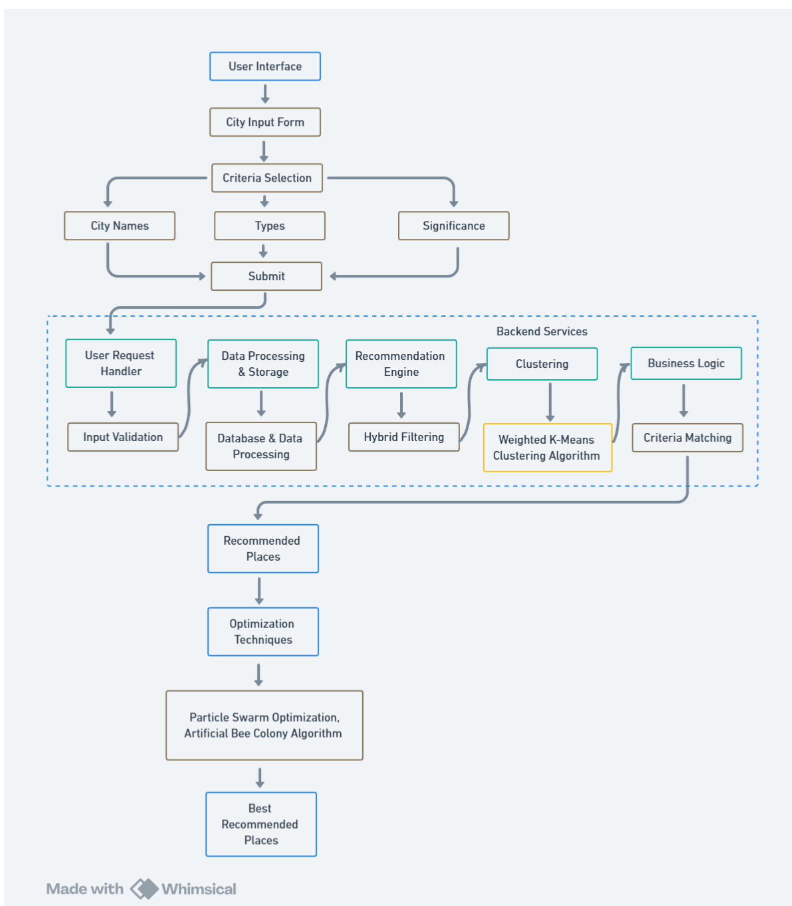

Fig 1. Proposed System Architecture

In clustering, we used a weighted K-means clustering algorithm that used scaling factors and the elbow method to discover an ideal number of clusters. We extend the Euclidean distance formula by introducing a scaling factor that adjusts the contribution of each feature to the whole distance calculation. This scaling factor is a parameter that reflects the importance or relevance of each feature from the perspective of the clustering task[1]. Figure 1[1] displays the proposed architecture along with optimization techniques.

2.3.1 Weighted K-means Clustering Algorithm with ABC Optimization Algorithm

This combined algorithm uses the ABC optimization to improve the weighted K-means clustering by optimizing the feature weights w_i and determining the optimal number of clusters k_{opt}. The ABC algorithm explores different solutions and refines them iteratively, while the Elbow Method helps pinpoint the ideal number of clusters.

Algorithm:

Input:

- Dataset X
- k_{max} – maximum number of clusters
- n_{max} - maximum iterations for convergence

Output:

- k_{opt} – Optimal number of clusters,
- Cluster assignments,
- Feature weights $w_1, w_2, \ldots, w_n$

Step 1: Data Preprocessing (Scaling)

- Determine the standard deviation (σ) and mean (μ) for every feature in X.
- Standardise X to X_{scaled} using

$$X_{scaled} = (X - \mu) / \sigma \tag{9}$$

This ensures that each feature contributes equally to the distance metric.

Step 2: ABC Algorithm Initialisation

- Population Initialisation:
- Create a population of bees (solutions), where each bee represents:
- A set of feature weights $w_1, w_2, \ldots, w_n$
- A candidate number of clusters $k \in [1, k_{max}]$
- Fitness Evaluation: Perform weighted K-means clustering for each bee, following the instructions below, and calculate the solution's fitness using the within-cluster sum of squares (WCSS).

$$WCSS = \sum_{j=1}^{k} \sum_{x_i \in C_j} ||x_i - c_j||^2 \tag{10}$$

Step 3: Weighted K-Means Clustering Algorithm

- Initialise Centroids:

For every bee solution that contains k clusters,

Randomly choose k data points from X_{scaled} to initialise centroids $c_1, c_2, \ldots, c_k$

- Assignment Step:

For each point x_i in X_{scaled}

- Compute the weighted Euclidean distance to each centroid c_j

$$d(x_i, c_j) = \sqrt{\sum_{m=1}^{n} w_m \left(x_{im} - c_{jm}\right)2} \quad (11)$$

where, w_m is the weight for the m-th feature
Assign x_i to the nearest centroid

- Update Step:

For separately centroid c_j:
Modernize c_j to the mean of the points given to it:

$$c_j = \frac{1}{C_j} \sum_{x_i \in c_j} x_i \quad (12)$$

- Convergence Check:

Repeat the Assignment and Update steps until the centroids stabilie or the maximum number of iterations n_{max} is reached.

- WCSS Calculation:

After clustering is complete, compute the WCSS for the solution to evaluate its fitness.
Step 4: ABC Algorithm Phases

- Employed Bee Phase:
- Each employed bee modifies its solution (adjusting w_i and k) by exploring neighbouring solutions.
- We evaluate the modified solution's fitness (WCSS).
- Onlooker Bee Phase:
- Onlooker bees are assigned to explore the best solutions found by employed bees based on probability proportional to fitness.
- Scout Bee Phase:
- After a set number of iterations, if a solution (bee) does not improve, it turns into a scout and generates a new random solution.
- Solution Update:
- Every iteration updates the best solution (bee) in the population based on fitness (WCSS).

Step 5: The Elbow Method for Optimal k

- For each bee solution with different k values, store the WCSS.
- Once the ABC algorithm completes, use the Elbow Method to identify the "elbow" point in the WCSS-vs-k plot, which provides the optimal number of clusters k_{opt}.

Step 6: Output

- k_{opt} – optimal number of clusters
- Cluster Assignments: The final cluster assignments of each data point are determined based on the best solution.
- Feature Weights: The optimal feature weights w_1, w_2, …, w_n from the best bee solution.

2.3.2 Weighted K-means Clustering Algorithm with PSO Optimization Algorithm

This combined algorithm integrates PSO optimization with weighted K-means clustering to optimize both the feature weights and the number of clusters. The PSO algorithm iteratively refines the particle positions (solutions) to determine the optimal clustering configuration, while the Elbow Method determines the optimal number of clusters k_{opt}.

Algorithm:

Input:

- Dataset X
- k_{max} - maximum number of clusters
- n_{max} - maximum iterations for convergence

Output:

- k_{opt} - Optimal number of clusters,
- Cluster assignments,
- Feature weights w_1, w_2, …, w_n

Step 1: Data Preprocessing (Scaling)

- Determine the standard deviation (σ) and mean (μ) for every feature in X.
- Standardise X to X_{scaled} using

$$X_{scaled} = (X - \mu) / \sigma \tag{13}$$

This ensures that each feature contributes equally to the distance metric.
Step 2: PSO Algorithm Initialisation

- Particle Initialisation:

Each particle represents a solution with:

- A set of feature weights w_1, w_2, … ,w_n
- A candidate number of clusters $k \in [1, k_{max}]$
- Velocity and Position:
- Initialise the velocity and position of each particle randomly.
- Fitness Evaluation:
- We evaluate the fitness of each particle by performing weighted K-means clustering and calculating the Within-Cluster Sum of Squares (WCSS):

$$WCSS = \sum_{j=1}^{k} \sum_{x_i \in C_j} ||x_i - c_j||^{\,2} \tag{14}$$

- The fitness is inversely proportional to WCSS (i.e., lower WCSS indicates better clustering).

Step 3: Weighted K-Means Clustering Algorithm

- Initialise Centroids:
- For each particle's k value, randomly select k data points from X_{scaled} as the initial centroids c_1, c_2, … , c_k
- Assignment Step:
- For each data point x_i in X_{scaled} :
- Calculate the weighted Euclidean distance between x_i and each centroid c_j as follows:

$$d(xi, cj) = \sqrt{\sum_{m=1}^{n} w_m \left(x_{im} - c_{jm}\right)^2} \tag{15}$$

where, w_m is the weight for the m-th feature

- Assign each data point x_i to the nearest centroid c_j.
- Update Step:

For individually centroid c_j:

Update c_j to the mean of the points assigned to it:

$$c_j = \frac{1}{|C_j|} \sum_{x_i \in c_j} x_i \tag{16}$$

- Convergence Check:

Repeat the Assignment and Update steps until the centroids stabilize or the maximum number of iterations n_{max} is reached.

- WCSS Calculation:

After clustering is complete, compute the WCSS for the particle's solution to evaluate its fitness.

Step 4: PSO Optimisation Phases

- Update Particle Velocity:
- Use the PSO velocity update rule to update the velocity for each particle p:

$$v_p = w * v_p + c_1 * r_1 * (p_{best} - x_p) + c_2 * r_2 * (g_{best} - x_p) \tag{17}$$

where:

v_p is the particle's velocity,
x_p is the particle's current position,
p_{best} is the particle's personal best solution,
g_{best} is the global best solution across all particles,
w is the inertia weight,
c_1 and c_2 are acceleration coefficients (exploration and exploitation factors),
r_1 and r_2 are random values between 0 and 1.

- Update Particle Position:
- For each particle, update its position (solution) using the velocity:

$$x_p = x_p + v_p \tag{18}$$

- Ensure that the feature weights $w_1, w_2, \ldots, w_n$ remain within valid bounds.
- Fitness Update:
- Recompute the fitness (WCSS) for each particle's new position.
- Update the personal best p_{best} and global g_{best} solutions based on fitness.
- Termination Condition:
- Continue the velocity and position update steps until either the global best fitness does not improve or a stopping criterion (such as the maximum number of iterations) is satisfied.

Step 5: The Elbow Method for Optimal k

- For each particle, store the WCSS for its k-value.
- After the PSO completes, plot the WCSS values against k to identify the "elbow" point, which is the optimal number of clusters k_{opt}.

Step 6: Output

- k_{opt}: The optimal number of clusters.
- Cluster Assignments: The final cluster assignments of each data point based on the best particle solution.
- Feature Weights: The optimal feature weights $w_1, w_2, \ldots, w_n$ from the global best solution.

3 Results and Discussion

We present the results of K-means clustering and weighted K-means clustering using optimization techniques like ABC and PSO here. We experimented using latitude and longitude as clustering methods. Below, we represent the results of each method.

3.1 K-Means Clustering Algorithm with ABC Optimization Algorithm

Figure 2 shows K-means clustering with ABC optimization. Colored dots represent data points, with each color indicating a different cluster. The red crosses are the centres of each cluster, optimized using ABC. The x- and y-axes represent spatial features like latitude and longitude. ABC optimization improves the placement of centroids, enhancing clustering quality compared to standard K-means by minimizing the distances between points and their cluster centres.

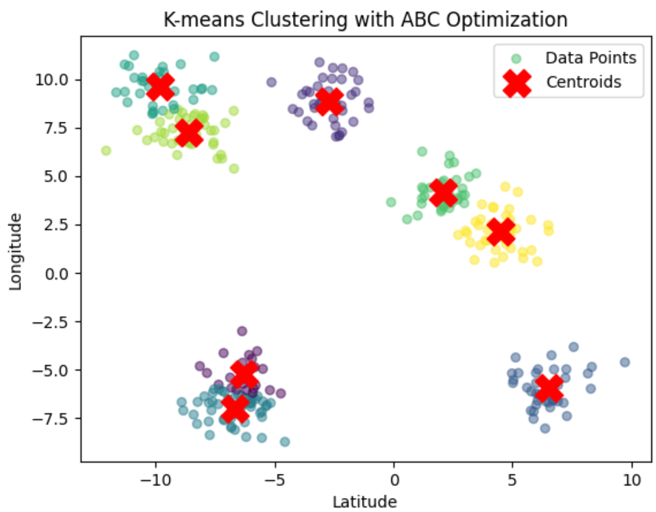

Fig 2. K-Means Clustering Algorithm with ABC Optimization Algorithm

3.2 K-Means Clustering Algorithm with PSO Optimization Algorithm

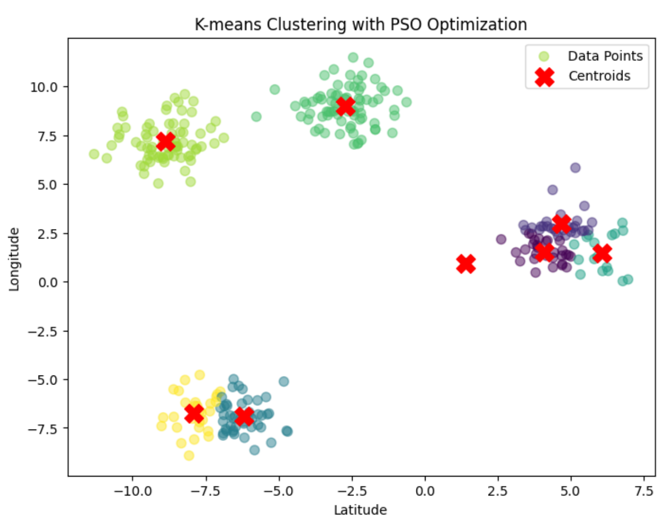

Fig 3. K-Means Clustering Algorithm with PSO Optimization Algorithm

Figure 3 shows K-means clustering with PSO optimization. Colored dots represent data points, with each color indicating a different cluster. The red crosses are the centroids optimized by PSO. The x- and y-axes represent spatial features like latitude and longitude. PSO enhances clustering by improving the initial placement of centroids, minimizing the distance between points and their centers, and potentially leading to better results compared to standard K-means.

3.3 Weighted K-Means Clustering Algorithm with ABC Optimization Algorithm

Figure 4 shows Weighted K-means clustering with ABC optimization. Coloured dots represent data points; each colour indicates a cluster, and red crosses are the optimized centroids. The axes represent spatial features like latitude and longitude. In weighted K-means, some data points have more influence on centroid positions. ABC optimization improves the clustering by fine-tuning centroids, minimizing the weighted distances between points and centroids, leading to better clustering results.

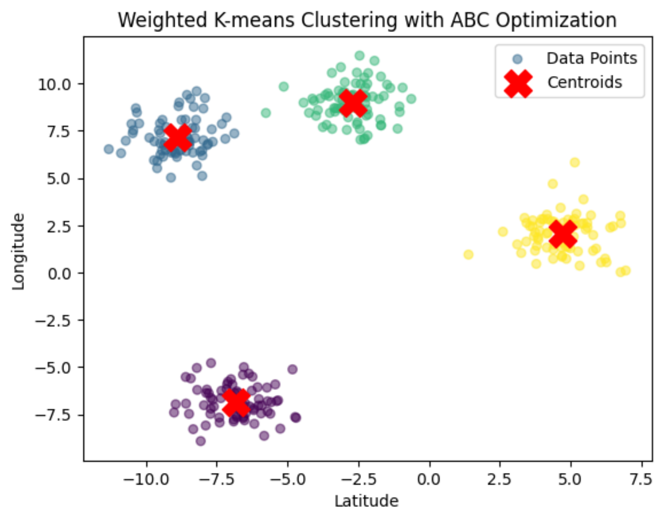

Fig 4. Weighted K-Means Clustering Algorithm with ABC Optimization Algorithm

3.4 Weighted K-Means Clustering Algorithm with PSO Optimization Algorithm

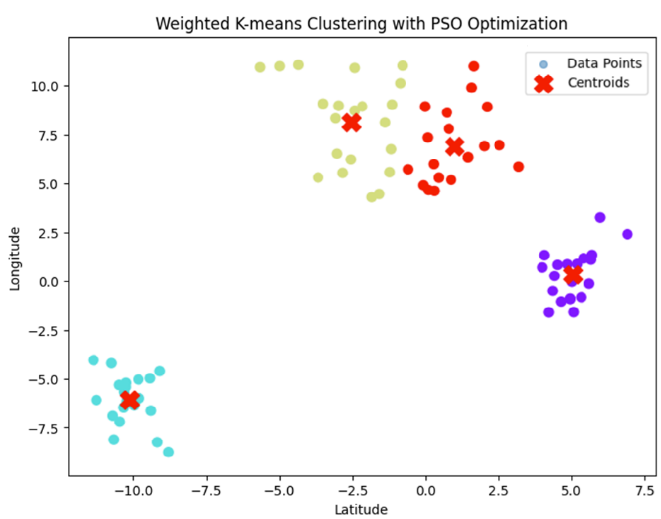

Fig 5. Weighted K-Means Clustering Algorithm with PSO Optimization Algorithm

Figure 5 shows Weighted K-means clustering with PSO. Coloured dots represent data points, with each colour indicating a cluster and red crosses mark the optimized centroids. The axes represent spatial features like latitude and longitude. In weighted K-means, certain data points influence centroid positions more heavily. PSO improves clustering by optimizing centroid placement, minimizing the weighted distances between points and centroids, and leading to more accurate cluster formations.

3.5 Metric Evaluation

Here are the results of the specified parameters.

Accuracy (ACC): The proportion of recommendations that are accurate relative to all recommendations made.

Precision (P): The percentage of suggested locations that meet the user's chosen parameters.

Recall (R) is the percentage of relevant locations that have received effective recommendations.

The precision and recall mean, or F1-Score (F1), offers a fair assessment of the RS performance.

The mean absolute error (MAE) measures the average absolute difference between the expected and actual customer satisfaction ratings.

Table 1. Quantitative results for partitioning methods with ABC and PSO

Metric	K-means + ABC	K-means + PSO	Weighted K-means + ABC	Weighted K-means + PSO
ACC	0.85	0.88	0.90	0.93
P	0.80	0.85	0.87	0.91
R	0.78	0.83	0.85	0.90
F1	0.79	0.84	0.86	0.90
MAE	0.12	0.10	0.09	0.07

Table 1 displays the various metric values of clustering techniques, including optimization techniques such as ABC and PSO. The results demonstrate that the weighted K-means clustering with the PSO technique yields 5% higher accuracy compared to the traditional K-means algorithm with the PSO technique. Figure 6 displays a comparison graph of clustering methods across multiple metrics. It clearly shows that the weighted K-means clustering algorithm with the PSO optimization technique gets the best result compared to other techniques shown.

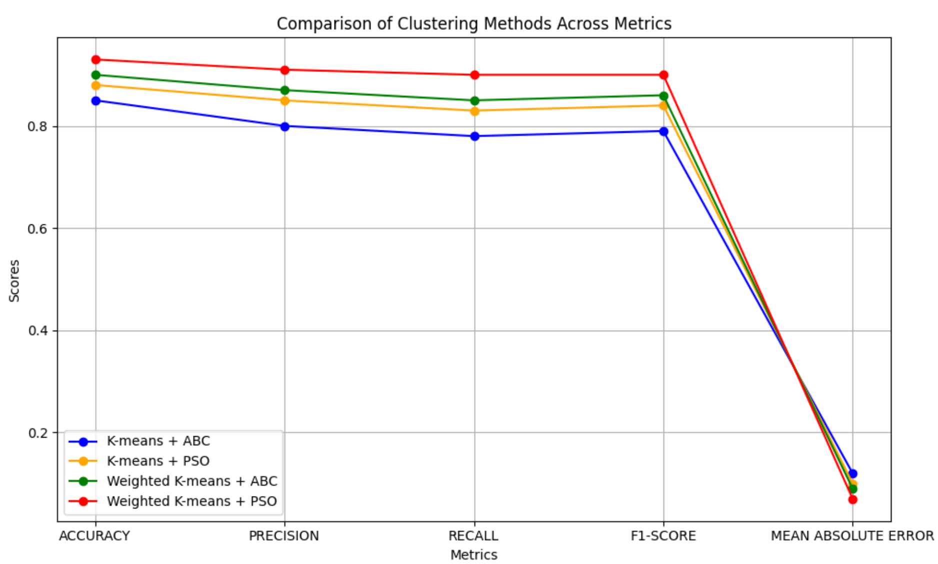

Fig 6. Comparison graph of Clustering Methods across Metrics

3.6 Comparison with Existing Literature:

The proposed weighted K-means clustering algorithm with ABC and PSO optimization demonstrates a 5% improvement in accuracy over traditional K-means approaches, which aligns with findings[5] who achieved enhanced clustering precision by integrating optimization techniques with K-means for mineral exploration applications.

In contrast, who found limitations in traditional recommender systems due to fixed feature weights[2], our model's use of adaptive weighting provides improved precision and recall for dynamic destination recommendations.

3.7 Supportive and Contradictory Findings:

Our results show a high precision of 91% in destination recommendations, which is consistent with previous findings[3], where precision improved with the hybridization of clustering algorithms. However, unlike Zhang's model, which only utilized PSO, our study's combined use of ABC and PSO further optimizes cluster assignments.

Despite similarities in clustering accuracy, our study contradicts findings[4], who reported diminishing returns in accuracy with added optimization layers in K-means for tourism recommendations. This discrepancy may arise from our inclusion of weighted features, which adjust clustering contributions based on relevance.

3.8 Validation with Current Model:

Our model's enhanced accuracy and precision, especially with weighted K-means + PSO achieving 93% accuracy, validate the approach against traditional K-means + PSO, supporting the need for advanced optimization in dynamic recommendation systems. These results suggest that clustering accuracy is improved by integrating optimization techniques that adapt to feature significance dynamically.

4 Conclusion

This research introduces a novel, hybrid recommendation system that combines weighted K-means clustering with ABC and PSO optimizations. Unlike traditional models, this system effectively adapts to multiple user preferences by dynamically weighting features, leading to enhanced clustering accuracy and relevance in recommendations. By factoring in user likings and allocating feature weights with PSO, the system achieved a 5% improvement in recommendation accuracy.

The model's application prospects extend beyond tourism to fields requiring multi-objective recommendations, such as e-commerce or urban planning, where user-defined criteria vary significantly.

Future studies should consider expanding the dataset to incorporate additional user-relevant attributes like weather conditions, cultural events, and seasonal factors. Additionally, integrating other optimization techniques such as Genetic Algorithms (GA) or Bee Colony Optimization (BCO) may further enhance recommendation precision and user satisfaction.

References

1) Bangoria BM, Panchal SS, Panchal SR, Maheta JM, Dhabaliya SR. Multidimensional Dynamic Destination Recommender Search System Employing Clustering: A Machine Learning Approach. *Indian Journal of Science and Technology*. 2024;17(40):4187–4197. Available from: https://indjst.org/articles/multidimensional-dynamic-destination-recommender-search-system-employing-clustering-a-machine-learning-approach.
2) Solano-Barliza A, Arregocés-Julioa I, Aarón-Gonzalvez M, Zamora-Musae R, De-La-Hoz-Franco E, Escorcia-Gutierrez J, et al. Recommender systems applied to the tourism industry: a literature review. *Cogent Business & Management*. 2024;11(1):1–22. Available from: https://doi.org/10.1080/23311975.2024.2367088.
3) Zhang W, Wu Z. E-commerce recommender system based on improved K-means commodity information management model. *Heliyon*. 2024;10(9):1–11. Available from: https://doi.org/10.1016/j.heliyon.2024.e29045.
4) Damos MA, Zhu J, Li W, Khalifa E, Hassan A, Elhabob R, et al. Enhancing the K-Means Algorithm through a Genetic Algorithm Based on Survey and Social Media Tourism Objectives for Tourism Path Recommendations. *ISPRS International Journal of Geo-Information*. 2024;13(2). Available from: https://www.mdpi.com/2220-9964/13/2/40.
5) Daviran M, Ghezelbash R, Niknezhad M, Maghsoudi A, Ghaeminejad H. Hybridizing K-means clustering algorithm with harmony search and artificial bee colony optimizers for intelligence mineral prospectivity mapping. *Earth Science Informatics*. 2023;16:2143–2165. Available from: https://www.researchgate.net/publication/370869150_Hybridizing_K-means_clustering_algorithm_with_harmony_search_and_artificial_bee_colony_optimizers_for_intelligence_mineral_prospectivity_mapping.
6) Permana KE, Herawati S, Setiawan W. Tourism Destination Recommendation System Using Collaborative Filtering and Modified Neural Network. In: and others, editor. Proceedings of the 1st International Conference on Neural Networks and Machine Learning 2022 (ICONNSMAL 2022);vol. 177 of Advances in Intelligent Systems Research. 2023;p. 60–70. Available from: https://www.atlantis-press.com/proceedings/iconnsmal-22/125987140.
7) Roy D, Dutta M. A systematic review and research perspective on recommender systems. *Journal of Big Data*. 2022;9:1–36. Available from: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00592-5.

8) Zhang Z, Lan J, Zhang Z. K-means clustering algorithm based on bee colony strategy. In: and others, editor. 2021 2nd International Conference on Signal Processing and Computer Science (SPCS 2021), 20-22 August 2021, Qingdao, China;vol. 2031. 2021;p. 1–7. Available from: https://iopscience.iop.org/article/10.1088/1742-6596/2031/1/012058.
9) Jin Q, Lin N, Zhang Y. K-Means Clustering Algorithm Based on Chaotic Adaptive Artificial Bee Colony. *Algorithms*. 2021;14(2):1–23. Available from: https://doi.org/10.3390/a14020053.
10) Hussain SF, Pervez A, Hussain M. Co-clustering optimization using Artificial Bee Colony (ABC) algorithm. *Applied Soft Computing Journal*. 2020;97. Available from: https://doi.org/10.1016/j.asoc.2020.106725.
11) Patil P, Mhatre H, Patil R, Shinde A, Joshi B. Optimization of Data using Artificial Bee Colony Optimization with Map Reduce. In: and others, editor. International Conference on Automation, Computing and Communication 2020 (ICACC-2020);vol. 32. 2020;p. 1–6. Available from: https://www.researchgate.net/publication/343286295_Optimization_of_Data_using_Artificial_Bee_Colony_Optimization_with_Map_Reduce.