International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Application of Machine Learning in Movie Recommendation using Harris Hawks Optimization and K-means (HHO-k-means) Clustering

Ambikesh G[1*], S. Shrikantha Rao [2], K. Chandrasekaran[3]

Submitted: 21/03/2023 **Revised**: 25/05/2023 **Accepted**: 11/06/2023

Abstract: In this study, a novel movie recommender system with Harris Hawks Optimization— k-means (HHO-k-means) clustering is proposed. The paper presents an empirical comparison of several clustering algorithms - k-means, PCA-k-means, SOM-Cluster, PCA-SOM, and HHO-k-means - across varying numbers of clusters. The performance metrics employed are Precision, Recall, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The results show that the HHO-k-means algorithm consistently outperforms the other methods in terms of these metrics across all cluster sizes. It demonstrates higher precision, higher recall, lower MAE, and lower RMSE. Conversely, the PCA-k-means method generally exhibits less favorable results as the number of clusters increases. These findings suggest that the HHO-k-means algorithm may provide a more accurate clustering approach.

Keywords*: Clustering Algorithms, k-means, PCA-k-means, SOM-Cluster, PCA-SOM, HHO-k-means, Recall, Mean Absolute Error, Root Mean Square Error.*

1. Introduction

In recent years, the growth of digital platforms like Netflix, Amazon Prime, and Hulu has led to an explosion in the amount of available content. With such an extensive range of movies and TV shows, users often find it challenging to discover new content that aligns with their preferences. To help users navigate through this vast ocean of content, movie recommender systems have become increasingly popular [1]. These systems use a combination of machine learning techniques and algorithms to provide users with personalized recommendations based on their viewing history, preferences, and other relevant information [2].

Movie recommender systems are designed to filter, analyze, and rank content based on a user's preferences, providing personalized suggestions for movies and TV shows. These systems are a critical component of modern streaming platforms, enabling users to discover new content tailored to their interests quickly [3].

There are three primary types of movie recommender systems: content-based filtering, collaborative filtering, and hybrid filtering [4]. Content-based filtering focuses on the features of the content itself, such as genre, director, and actors, to generate recommendations [5]. Collaborative filtering, on the other hand, relies on the user's interactions with the content and similarities between users to make recommendations. Hybrid filtering combines both content-based and collaborative filtering techniques, ensuring a more comprehensive and accurate recommendation process [6].

In recent years, researchers have explored various optimization algorithms [7] [8] to improve the performance of machine learning techniques [9] [10]. Several such attempts have been made to enhance the performance of movie recommender systems. Peng and Gong [11] demonstrated a focus on refining the accuracy of recommendation systems by optimizing collaborative filtering algorithms. They strived to improve similarity calculations between movie attributes. Their endeavor to enhance accuracy is mirrored in a study by Wang et al. [12] that brought together Support Vector Machine (SVM) and an advanced Particle Swarm Optimization (PSO) algorithm. In their model, elements such as item content, user demographics, and behavioral data were incorporated to attain better precision. This drive towards precision and accuracy in recommendations has led to novel methods such as the sentiment analysis-based system developed by Roy and Dutta [13]. By using the water cycle earthworm Optimization (WCEWO) technique, they achieved an impressive maximum accuracy of 89.81%. Similarly, Yang and Duan [14] merged manifold learning and ensemble learning while Zhou et al. [15] employed an evolutionary search for dynamic multiobjective optimization problems, both to optimize the recommendation model and enhance prediction accuracy.

[1, 2] Department of Mechanical Engineering, National Institute of Technology Karnataka, India 575025
[3]Department of Computer Science and Engineering, National Institute of Technology Karnataka, India 575025
[1]ambig45@gmail.com, [2]Ssrcsr@gmail.com, [3]kch@nitk.edu.in
**Corresponding author: Ambikesh G, ambig45@gmail.com*

In the quest for improved performance and precision, hybrid models have been particularly impactful. Katarya and Verma [16] showcased this by using K-means and PSO to optimize the fuzzy c-means in a collaborative movie recommendation system. Other hybrid recommender systems, like the one presented by Katarya [17], combined the k-means clustering algorithm with a bio-inspired artificial bee colony optimization technique, while Zhixiang [18] incorporated a time-based interest calculation to refine the recommendation accuracy.

The field of sentiment analysis also holds significant relevance. Banerjee et al. [19] underscored the role of sentiment analysis in decision-making and its increasing popularity in computer science. A testament to its utility in recommendations was Srinivasarao et al.'s [20] sentiment-based movie recommendation system using the SVM Classifier combined with the Harris Hawks optimization method, demonstrating robust performance on high-dimensional data with a commendable 97% accuracy.

However, recommendation systems are not without their challenges, such as data sparsity and cold-start problems. In this context, Parthasarathy and Kalivaradhan [21] utilized a density-based clustering method combined with artificial flora and content-boosted collaborative filtering to address data sparsity issues effectively. The cold-start problem was tackled by Liang et al. [22], who developed a weight-normalized movie recommendation model, demonstrating improved recommendation accuracy.

Geng et al. [23] amalgamated multi-objective optimization algorithm with a recommendation algorithm to create a multi-objective hydrologic cycle optimization. Likewise, Sridhar et al. [24] innovatively proposed a hybrid movie recommendation model using monarch butterfly optimization and deep belief network. Addressing the speed, scalability, and cold start issues, Sandeep and Prabhu [25] introduced a collaborative movie recommendation system combining K-means clustering with an Ant Colony Optimization technique. In a similar vein, Sharma et al. [26] proposed a Firefly clustering technique to optimize movie recommendation results, delivering superior performance compared to traditional algorithms like K-means and fuzzy C-means. Investigations into diversity and novelty in recommendations led Keat et al. [27] to implement Deep Reinforcement Learning approaches, resulting in superior results in these parameters compared to a probabilistic-based multi-objective approach using evolutionary algorithms. In contrast, Almeida and Britto [28] employed a Multi-objective Evolutionary Algorithm in a content-based recommendation system aiming to address the limitations of single-criterion recommendations.

Movie recommendation systems have also been shown to benefit significantly from hybridization with optimization techniques. A testament to this was the system introduced by Mohapatra et al. [29] that utilized k-means clustering and a cuckoo search optimization algorithm. Furthermore, Vellaichamy and Kalimuthu [30] proposed a hybrid Collaborative Movie Recommender system combining Fuzzy C Means clustering with Bat optimization, aimed at resolving scalability problems and improving recommendation quality. Several researchers also devoted attention to temporal dynamics and adaptability in recommendations. Zhang and Mao [31] introduced a model termed Markovian factorization of matrix process that addressed these aspects in collaborative filtering problems. Similarly, Chinthareddy et al. [32] used computational intelligence and cuckoo search optimization to obtain optimal weights from similarity metric weights, which led to a significant reduction in prediction error.

A few studies considered user-specific factors and security issues. For instance, Dooms et al. [33] designed a self-learning, user-specific hybrid recommender system that balanced responsiveness, scalability, system transparency, and user control. Meanwhile, Verma and Dixit [34] proposed a hybrid model that integrated Entropy-Based Mean clustering and PSO techniques to secure movie recommendations from shilling attacks.

While considerable strides have been made in improving the precision and accuracy of movie recommendation systems through various optimization algorithms and hybrid models, research gaps still exist. Most significantly, the need to further enhance the scalability and performance of these systems and tackle inherent issues such as data sparsity and cold-start problems remains. Moreover, the implementation of novel techniques such as the proposed Harris Hawks Optimization— K-means (HHO-k-means) Clustering could offer fresh perspectives and potential solutions in this dynamic field. The integration of other emerging techniques, such as deep reinforcement learning and advanced sentiment analysis, could further diversify and improve the landscape of movie recommendation systems. Thus, in this paper, Harris Hawks Optimization (HHO) algorithm is combined with K-means clustering to develop a more efficient and accurate movie recommender system.

2. Methodology

2.1. Harris Hawks Optimization Algorithm

HHO algorithm is a nature-inspired optimization algorithm based on the hunting behavior of Harris Hawks [35]. This algorithm simulates the Harris Hawks' cooperative hunting strategy, consisting of three main phases: exploration, exploitation, and encircling the prey.

During the exploration phase, the algorithm initializes a random set of solutions, also known as hawks, and updates their positions after each iteration. This phase allows the HHO algorithm to perform a global search for the optimal solution.

In the exploitation phase, the hawks encircle the prey by updating their positions toward the best solution found during exploration. This phase focuses on refining the search space and converging to the optimal solution.

Finally, the encircling phase involves the hawks converging on the prey by updating their positions using a combination of shrinking encircling and spiral updating mechanisms. The shrinking encircling mechanism gradually narrows the search space, while the spiral updating mechanism ensures that the algorithm converges to the optimal solution. The pseudo code of HHO is given in Algorithm 1.

Algorithm 1: Harris Hawks Optimization (HHO) algorithm

```
Input: Number of Harris Hawks (N), Number of Iterations (I)
Objective Function (F), Lower Bound (LB), Upper Bound (UB)
Initialize the positions of hawks
For each iteration do:
    Calculate the energy
    Find the best hawk with maximum E
    For each hawk do:
        Generate a random number (r)
        If r >= 0.5 then
            If E > 1 then
                Perform Soft besiege state
            Else
                Perform Hard besiege state with progressive rapid dive
            End if
        Else
        Perform exploration state
        End if
        Update the position of hawk
    End for
End for
Return: The best solution and the optimal value
```

2.2. K-means Clustering

K-means clustering is a widely-used unsupervised machine learning technique that aims to partition a dataset into K clusters based on similarity. In the context of movie recommender systems, K-means clustering is employed to group movies with similar features and user preferences.

The algorithm works by randomly selecting K initial centroids (center of clusters) and iteratively updating their positions until convergence. The process involves calculating the distance between each data point and the centroids, assigning each data point to the closest centroid, and updating the centroid positions based on the average of the assigned data points. This process is repeated until the centroids' positions stabilize or the desired number of iterations is reached. The pseudo code of K-means Clustering is given in Algorithm 2.

Algorithm 2: K-means Clustering

Input: Data set (D), Number of clusters (K)

Randomly initialize K centroids

Repeat until convergence:

  Assign each data point to the closest centroid

  Recalculate centroid positions based on the mean of the points in each cluster

End repeat

Return: Centroid positions and cluster assignments for each point in D

2.3. Hybrid Approach: Harris Hawks Optimization - K-means Clustering

Combining the Harris Hawks Optimization algorithm with K-means clustering results in a more efficient and accurate movie recommender system. The HHO algorithm enhances the K-means clustering process by optimizing the initial centroids' selection and updating their positions more effectively.

This hybrid approach offers several benefits over traditional movie recommender systems:

- Improved Accuracy: The combination of HHO and K-means clustering provides a more accurate recommendation system, ensuring that users receive relevant content tailored to their preferences.
- Scalability: The hybrid approach can efficiently handle large datasets, making it suitable for modern streaming platforms with vast content libraries.
- Robustness: The HHO algorithm's exploration and exploitation phases ensure a more comprehensive search for the optimal solution, reducing the risk of getting stuck in local optima.
- Faster Convergence: The HHO algorithm's encircling mechanism, combined with K-means clustering, ensures faster convergence to the optimal solution, resulting in a more efficient recommendation process.

Algorithm 3: Harris Hawks Optimization algorithm with K-means clustering

Input: Data set (D), Number of clusters (K), Number of Harris Hawks (N), Number of Iterations (I), Lower Bound (LB), Upper Bound (UB)

Use HHO to find optimal initial centroids:

Initialize the positions of hawks

For each iteration do:

  Calculate the energy (E) of hawks

  Find the best hawk with maximum E

  For each hawk do:

    Generate a random number (r)

    If r >= 0.5 then

      If E > 1 then

        Perform Soft besiege state

      Else

        Perform Hard besiege state with progressive rapid dive

      End if

    Else

Perform exploration state

End if

Update the position of hawk

End for

End for

Use K-means with the optimized initial centroids:

Repeat until convergence:

Assign each data point to the closest centroid

Recalculate centroid positions based on the mean of the points in each cluster

End repeat

Return: Centroid positions and cluster assignments for each point in D

2.4. Evaluation and Performance Metrics

To assess the performance of a movie recommender system using the hybrid HHO-K-means clustering approach, several performance metrics can be employed. These metrics include:

Precision: Measures the proportion of relevant recommendations among the total recommendations made. A higher precision value indicates a more accurate and efficient system.

$$Precision = \frac{|interesting \cap TopN|}{N}$$

Recall: Calculates the proportion of relevant recommendations found among all possible relevant items. A higher recall value suggests that the system effectively identifies relevant content for users.

$$Recall = \frac{|interesting \cap TopN|}{|interesting|}$$

Mean Absolute Error (MAE): Represents the average difference between the predicted and actual values. A lower MAE value indicates a more accurate recommendation system.

$$MAE = \frac{\sum_{(i,j)}|P_{ij} - R_{ij}|}{M}$$

where MAE depicts the sum of the difference between the expected rating P_{ij} and the actual rating r_{ij} over movies quantity M.

Computational Time: It measures the time taken by the algorithm to generate recommendations [36]. A shorter computational time implies a faster and more efficient system [37].

2.5. Dataset Description

The data used in the study was movies data from the MovieLens website, which offers movie recommendation service [38]. It was collected and maintained by a research group of the University of Minnesota called GroupLens. The 1M dataset contain only the demographic data, ratings and movie data [38] [39]. This stable benchmark dataset has 100000 ratings of 1700 movies from 1000 users. It was released in April 1998. It can be found and downloaded for open-source use at https://grouplens.org/datasets/movielens/100k/.

3. Results and Discussions

3.1. Precision

In Table 1, the proposed HHO-k-means algorithm outperforms the other four algorithms (traditional k-means, PCA-k-means, SOM-Cluster, PCA-SOM) in all the cases for different numbers of clusters. This suggests that the HHO-k-means algorithm is more precise and potentially more reliable for movie recommendation. Comparing the results between different cluster sizes, the highest precision for the HHO-k-means algorithm is achieved at 6 clusters (0.5619). For traditional k-means, the highest precision is also achieved at 6 clusters, but the value is significantly lower (0.1352). This same trend is observed for PCA-k-means, with the highest precision (0.1886) being achieved at 6 clusters. On the other hand, SOM-Cluster and PCA-SOM algorithms showed a different trend where the precision values do not consistently increase or decrease with the number of clusters. The highest precision for the SOM-Cluster (0.3573) and PCA-SOM (0.4247) algorithms are achieved at 6 clusters. The performance of traditional k-means and PCA-k-means decreases after reaching a peak at 6 clusters, while the precision of HHO-k-means decreases slightly after 6 clusters but remains relatively stable (Figure 1). This stability in precision with the increase in the number of clusters suggests that the HHO-k-means algorithm can maintain a relatively high level of accuracy, even with more complex clustering scenarios. It is worth noting that the combination of PCA-SOM, while more effective than standalone traditional k-means and PCA-k-means, is still less precise than HHO-k-means. This suggests that the application of the Harris Hawks Optimization to k-means clustering yields a notable improvement in the precision of movie recommendations. This higher precision of HHO-k-means algorithm can lead to better recommendations, enhancing the user experience by offering movies that align more closely with their preferences.

Table 1. Precision of various algorithms for various number of clusters

No. of clusters	k-means	PCA-k-means	SOM-Cluster	PCA-SOM	HHO-k-means
3	0.1101	0.1415	0.3192	0.3554	0.4697
6	0.1352	0.1886	0.3573	0.4247	0.5619
9	0.1211	0.1855	0.3554	0.4063	0.5271
12	0.112	0.1644	0.3251	0.3651	0.5039
15	0.1192	0.1811	0.3339	0.3663	0.5151

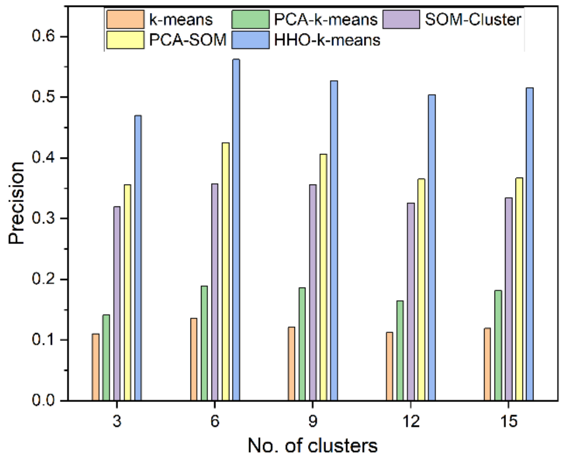


Fig 1. Precision of proposed HHO-k-means for various number of clusters

3.2. Recall

Table 2 provides a comparison of recall values across different clustering algorithms with varying numbers of clusters. The HHO-k-means algorithm consistently exhibits the highest recall among all the methods across varying numbers of clusters. The highest recall for HHO-k-means is 0.474696, observed with 6 clusters. The PCA-SOM method shows a fairly consistent recall but fluctuates with the change in the number of clusters. The highest recall for PCA-SOM is observed with 6 clusters (0.2839). The SOM-Cluster and PCA-k-means methods generally show an increase in recall as the number of clusters increases from 3 to 12, but not consistently. The highest recall for SOM-Cluster is 0.2751 with 15 clusters, while the highest recall for PCA-k-means is 0.1401 with 12 clusters. The k-means algorithm has the lowest recall values across all numbers of clusters. Its highest recall is 0.0757, observed with 6 clusters. It should be noted that recall is a metric for classification problems that measures the ability of a method to find all the relevant cases within a dataset. The higher the recall, the better the method is at identifying positive instances in the data. These results suggest that the HHO-k-means algorithm outperforms the others in this regard (Figure 2).

Table 2. Recall of various algorithms for various number of clusters

No. of clusters	k-means	PCA-k-means	SOM-Cluster	PCA-SOM	HHO-k-means
3	0.0565	0.1262	0.1303	0.1581	0.3702
6	0.0757	0.1179	0.1817	0.2839	0.4747

9	0.0459	0.0976	0.1459	0.1917	0.2725
12	0.0597	0.1401	0.192	0.1802	0.3787
15	0.0545	0.0719	0.2751	0.1357	0.3603

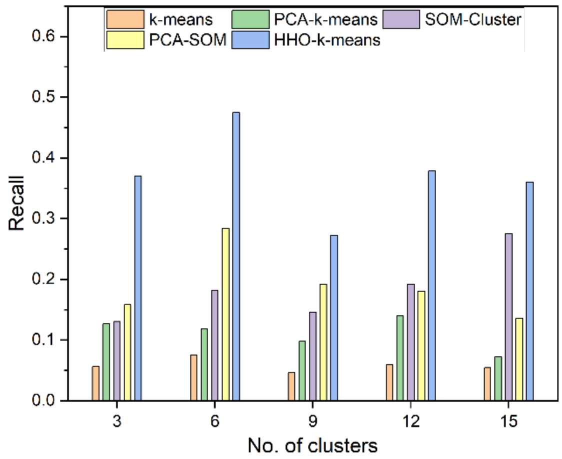


Figure 2. Recall of proposed HHO-k-means for various number of clusters

3.3. MAE

Table 3 presents a comparison of the Mean Absolute Error (MAE) produced by different clustering algorithms for varying numbers of clusters. The MAE is a measure of prediction error for numeric prediction models, with lower values indicating better performance. The HHO-k-means algorithm consistently produces the lowest MAE across all the various cluster sizes, indicating that this method is the most accurate in terms of average absolute deviation from the true values. The lowest MAE for HHO-k-means is observed with 6 clusters, where the MAE is approximately 0.6285. The PCA-k-means method generally exhibits a trend of increasing MAE with a rise in the number of clusters, suggesting a decrease in prediction accuracy. The lowest MAE for PCA-k-means is noted with 12 clusters, where the MAE is approximately 0.7655. For k-means, SOM-Cluster, and PCA-SOM algorithms, the MAE values fluctuate as the number of clusters varies, without showing a clear pattern. Among these, the lowest MAE is observed with SOM-Cluster for 6 clusters (MAE of approximately 0.7279), followed closely by the PCA-SOM method for 3 clusters (MAE of approximately 0.7529), and then the k-means method for 6 clusters (MAE of approximately 0.7952). The PCA-k-means method, with 15 clusters, exhibits the highest MAE (approximately 0.8787) across all methods and cluster numbers, indicating the least accurate. From these observations, it appears that the HHO-k-means algorithm performs the best in terms of minimizing the mean absolute error across varying numbers of clusters. Conversely, PCA-k-means, especially with a higher number of clusters, tends to be less accurate on average (Figure 3).

Table 3. MAE of various algorithms for various number of clusters

No. of clusters	k-means	PCA-k-means	SOM-Cluster	PCA-SOM	HHO-k-means
3	0.8286	0.8424	0.8197	0.7529	0.7133
6	0.7952	0.852	0.7279	0.7562	0.6285
9	0.8339	0.8216	0.8192	0.8004	0.6980
12	0.8096	0.7655	0.7864	0.8285	0.7002
15	0.8103	0.8787	0.7653	0.7884	0.6915

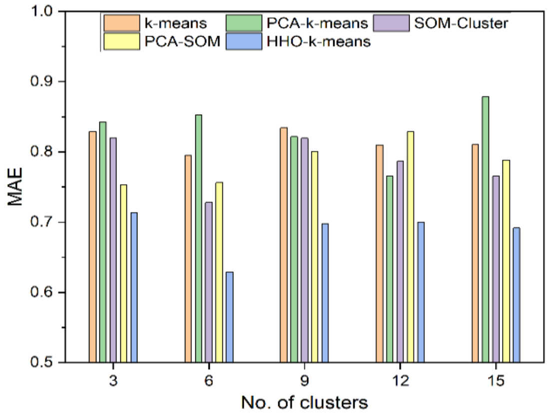


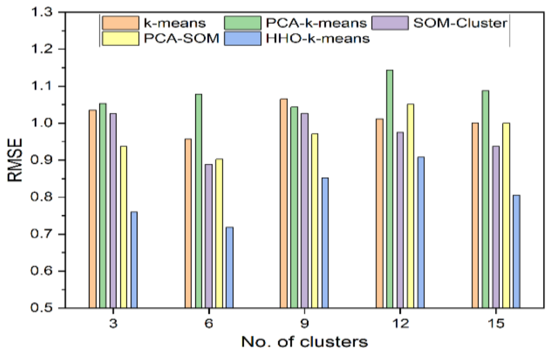
Fig 3. MAE of proposed HHO-k-means for various number of clusters

3.4. RMSE

Table 4 outlines the Root Mean Square Error (RMSE) of various clustering algorithms for a variety of cluster counts. The RMSE is a widely used measure of the differences between values predicted by a model and the values observed. It's particularly useful when large errors are notably undesirable, and a lower RMSE signifies better model performance. The HHO-k-means consistently has the lowest RMSE across all cluster counts, demonstrating that this method consistently provides the most accurate predictions with the smallest average error magnitude. The lowest RMSE for HHO-k-means is achieved with 6 clusters, which delivers an RMSE of approximately 0.7184. The PCA-k-means algorithm has a generally increasing RMSE as the cluster count grows, suggesting that its prediction accuracy deteriorates as more clusters are introduced. Its lowest RMSE is at 3 clusters, with a value of approximately 1.0532. The k-means, SOM-Cluster, and PCA-SOM algorithms do not display a clear pattern in RMSE as the number of clusters changes. The lowest RMSE for these methods occurs at 6 clusters for k-means (RMSE around 0.9571), 6 clusters for SOM-Cluster (RMSE around 0.8892), and 3 clusters for PCA-SOM (RMSE around 0.9369). The PCA-k-means, when applied to 12 clusters, results in the highest RMSE (around 1.1438) across all methods and cluster counts, indicating the least accurate predictions (Figure 4). Based on the RMSE values, the HHO-k-means algorithm demonstrates the best performance among the algorithms examined, with the PCA-k-means algorithm generally performing less well as the number of clusters increases.

Table 4. RMSE of various algorithms for various number of clusters

No. of clusters	k-means	PCA-k-means	SOM-Cluster	PCA-SOM	HHO-k-means
3	1.0357	1.0532	1.026	0.9369	0.7604
6	0.9571	1.0785	0.8892	0.9024	0.7184
9	1.0651	1.0439	1.0264	0.971	0.8525
12	1.0115	1.1438	0.9755	1.0515	0.9085
15	1.0008	1.0881	0.9381	0.9997	0.8058



Fige 4. RMSE of proposed HHO-k-means for various number of clusters

4. Conclusion

In this research, hybrid approach of Harris Hawks Optimization and K-means clustering is proposed for movie recommendation system. Five distinct clustering algorithms (k-means, PCA-k-means, SOM-Cluster, PCA-SOM, and HHO-k-means) are compared using various evaluation metrics. The results suggest that, on average, the HHO-k-means algorithm outperforms the other four methods across all cluster sizes. However, the PCA-k-means algorithm often yields less favorable results, particularly as the number of clusters increases. The hybrid approach of Harris Hawks Optimization and K-means clustering offers a promising solution for improving movie recommender systems' accuracy and efficiency. This approach effectively combines the strengths of both algorithms, resulting in a more accurate, scalable, robust, and efficient movie recommendation process.

Thus, it is recommended that further exploration and application of the HHO-k-means algorithm in various contexts, while considering the potential limitations of PCA-k-means as cluster count grows. Finally, it is crucial to note that while these results provide a broad comparison of these algorithms, the performance of any given algorithm may be data-specific, and individual application scenarios should be carefully assessed for the selection of the most appropriate clustering algorithm.

Future research could explore other optimization algorithms, deesp learning techniques, and data filtering methods to further enhance the performance of movie recommender systems. Additionally, integrating user demographics, social network data, and temporal information could provide a more comprehensive and personalized recommendation experience for users.

References

[1] Z. Yuan, J. H. Lee and S. Zhang. (2021). Optimization of the Hybrid Movie Recommendation System Based on Weighted Classification and User Collaborative Filtering Algorithm. Complexity, vol. 2021, 2021. doi: 10.1155/2021/4476560

[2] R. Katarya and O. P. Verma. (2017). An effective collaborative movie recommender system with cuckoo search. Egyptian Informatics Journal, vol. 18, p. 105 - 112, 2017. doi: 10.1016/j.eij.2016.10.002

[3] O. Kirmemis and A. Birturk. (2008). A content-based user model generation and optimization approach for movie recommendation.

[4] N. Ganesh, R. Shankar, R. Čep, S. Chakraborty and K. Kalita. (2023). Efficient feature selection using weighted superposition attraction optimization algorithm. Applied Sciences, vol. 13, p. 3223, 2023. doi: 10.3390/app13053223

[5] J. Priyadarshini, M. Premalatha, R. Čep, M. Jayasudha and K. Kalita. (2023). Analyzing Physics-Inspired Metaheuristic Algorithms in Feature Selection with K-Nearest-Neighbor. Applied Sciences, vol. 13, p. 906, 2023. doi: 10.3390/app13020906

[6] S. Papneja, K. Sharma and N. Khilwani. (2021). Movie recommendation to friends using whale optimization algorithm. Recent Advances in Computer Science and Communications, vol. 14, p. 1470 - 1475, 2021. doi: 10.2174/2213275912666190823104600

[7] N. Ganesh, R. Shankar, K. Kalita, P. Jangir, D. Oliva and M. Pérez-Cisneros. (2023). A Novel Decomposition-Based Multi-Objective Symbiotic Organism Search Optimization Algorithm. Mathematics, vol. 11, p. 1898, 2023. doi: 10.3390/math11081898

[8] S. Rajendran, N. Ganesh, R. Čep, R. C. Narayanan, S. Pal and K. Kalita. (2022). A Conceptual Comparison of Six Nature-Inspired Metaheuristic Algorithms in Process Optimization. Processes, vol. 10, p. 197, 2022. doi: 10.3390/pr10020197

[9] K. Shaik, J. V. N. Ramesh, M. Mahdal, M. Z. U. Rahman, S. Khasim and K. Kalita. (2023). Big Data Analytics Framework Using Squirrel Search Optimized Gradient Boosted Decision Tree for Heart Disease Diagnosis. Applied Sciences, vol. 13, p. 5236, 2023. doi: 10.3390/app13095236

[10] M. Joshi, K. Kalita, P. Jangir, I. Ahmadianfar and S. Chakraborty. (2023). A Conceptual Comparison of Dragonfly Algorithm Variants for CEC-2021 Global Optimization Problems. Arabian Journal for Science and Engineering, vol. 48, p. 1563-1593, 2023. doi: 10.1007/s13369-022-06880-9

[11] J. Peng and S. Gong. (2020). Optimization of Collaborative Filtering Algorithm in Movie Recommendation System. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12488 LNCS, p. 11 - 19, 2020. doi: 10.1007/978-3-030-62463-7_2

[12] Kapoor, E. ., Kumar, A. ., & Singh , D. . (2023). Energy-Efficient Flexible Flow Shop Scheduling With Due Date and Total Flow Time. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2s), 259–267. https://doi.org/10.17762/ijritcc.v11i2s.6145

[13] X. Wang, F. Luo, C. Sang, J. Zeng and S. Hirokawa. (2017). Personalized movie recommendation system based on support vector machine and improved particle swarm optimization. IEICE Transactions on

Information and Systems, vol. E100D, p. 285 - 293, 2017. doi: 10.1587/transinf.2016EDP7054
[14] D. Roy and M. Dutta. (2022). Optimal hierarchical attention network-based sentiment analysis for movie recommendation. Social Network Analysis and Mining, vol. 12, 2022. doi: 10.1007/s13278-022-00954-0
[15] K. Yang and Y. Duan. (2022). Personalized movie recommendation method based on ensemble learning. High Technology Letters, vol. 28, p. 56 - 62, 2022.
[16] W. Zhou, L. Feng, K. C. Tan, M. Jiang and Y. Liu. (2022). Evolutionary Search with Multiview Prediction for Dynamic Multiobjective Optimization. IEEE Transactions on Evolutionary Computation, vol. 26, p. 911 - 925, 2022. doi: 10.1109/TEVC.2021.3135020
[17] R. Katarya and O. P. Verma. (2016). A collaborative recommender system enhanced with particle swarm optimization technique. Multimedia Tools and Applications, vol. 75, p. 9225 - 9239, 2016. doi: 10.1007/s11042-016-3481-4
[18] R. Katarya. (2018). Movie recommender system with metaheuristic artificial bee. Neural Computing and Applications, vol. 30, p. 1983 - 1990, 2018. doi: 10.1007/s00521-017-3338-4
[19] L. Zhixiang. (2022). A Hybrid Movie Recommendation Algorithm Based on Optimized K-means Clustering. doi: 10.1109/IPEC54454.2022.9777625
[20] H. Banerjee, R. Dey, S. Chatterjee, A. Chakraborty, S. Pareek, S. Nayak, A. Gautam and H. N. Saha. (2017). Movie recommendation system using particle swarm optimization. doi: 10.1109/IEMECON.2017.8079574
[21] U. Srinivasarao, R. Karthikeyan, P. K. Sarangi and B. S. Panigrahi. (2022). Enhanced Movie Recommendation and Sentiment Analysis Model Achieved by Similarity Method through Cosine and Jaccard Similarity algorithms. 2022. doi: 10.1109/ICCCIS56430.2022.10037722
[22] J. Parthasarathy and R. B. Kalivaradhan. (2021). An effective content boosted collaborative filtering for movie recommendation systems using density based clustering with artificial flora optimization algorithm. International Journal of Systems Assurance Engineering and Management. doi: 10.1007/s13198-021-01101-2
[23] Z. Liang, Z. Yang and J. Cheng. (2023). Weight normalization optimization movie recommendation algorithm based on three-way neural interaction networks. Complex and Intelligent Systems, 2023. doi: 10.1007/s40747-023-01066-8
[24] S. Geng, C. Zhang, X. Yang and B. Niu. (2019). Multi-criteria recommender systems based on multi-objective hydrologic cycle optimization. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11656 LNCS, p. 92 - 102, 2019. doi: 10.1007/978-3-030-26354-6_9
[25] S. Sridhar, D. Dhanasekaran and G. C. P. Latha. (2023). Content-Based Movie Recommendation System Using MBO with DBN. Intelligent Automation and Soft Computing, vol. 35, p. 3241 - 3257, 2023. doi: 10.32604/iasc.2023.030361
[26] M. Sandeep Kumar and J. Prabhu. (2020). A hybrid model collaborative movie recommendation system using K-means clustering with ant colony optimisation. International Journal of Internet Technology and Secured Transactions, vol. 10, p. 337 - 354, 2020. doi: 10.1504/IJITST.2020.10028404
[27] Ms. Ritika Dhabalia, Ms. Kritika Dhabalia. (2012). An Intelligent Auto-Tracking Vehicle. International Journal of New Practices in Management and Engineering, 1(02), 08 - 13. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/5
[28] B. Sharma, S. K. Sharma, P. Bansal, N. S. Sushma and S. Sangam. (2022). Map-Reduce Based Parallel Firefly Algorithm for Fast Recommendations. 2022. doi: 10.1109/AIST55798.2022.10064743
[29] E. Y. Keat, N. M. Sharef, R. Yaakob, K. A. Kasmiran, E. Marlisah, N. Mustapha and M. Zolkepli. (2022). Multiobjective Deep Reinforcement Learning for Recommendation Systems. IEEE Access, vol. 10, p. 65011 - 65027, 2022. doi: 10.1109/ACCESS.2022.3181164
[30] M. S. Almeida and A. Britto. (2020). MOEA-RS: A Content-Based Recommendation System Supported by a Multi-objective Evolutionary Algorithm. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12416 LNAI, p. 265 - 276, 2020. doi: 10.1007/978-3-030-61534-5_24
[31] P. Mohapatra, R. K. Mohapatra and B. Sandhibigraha. (2019). Movie recommender system using improvised Cuckoo search. International Journal of Innovative Technology and Exploring Engineering, vol. 8, p. 2869 - 2873, 2019.
[32] V. Vellaichamy and V. Kalimuthu. (2017). Hybrid collaborative movie recommender system using clustering and bat optimization. International Journal of Intelligent Engineering and Systems, vol. 10, p. 38 - 47, 2017. doi: 10.22266/ijies2017.1031.05
[33] R. Zhang and Y. Mao. (2019). Movie Recommendation via Markovian Factorization of

Matrix Processes. IEEE Access, vol. 7, p. 13189 - 13199, 2019. doi: 10.1109/ACCESS.2019.2892289
[34] P. Chinthareddy, G. R. Nitta, S. V. Ramakrishna and D. Yakobu. (2017). Collaborative filtering recommendation system Cuckoo search optimization and K-nearest neighbor classifier. Journal of Advanced Research in Dynamical and Control Systems, vol. 9, p. 1217 - 1228, 2017.
[35] S. Dooms, T. De Pessemier and L. Martens. (2015). Online optimization for user-specific hybrid recommender systems. Multimedia Tools and Applications, vol. 74, p. 11297 - 11329, 2015. doi: 10.1007/s11042-014-2232-7
[36] A. K. Verma and V. S. Dixit. (2023). Collaborative filtering-based recommendations against shilling attacks with particle swarm optimiser and entropy-based mean clustering. International Journal of Information and Computer Security, vol. 20, p. 133 - 144, 2023. doi: 10.1504/IJICS.2023.10052969
[37] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja and H. Chen. (2019). Harris hawks optimization: Algorithm and applications. Future generation computer systems, vol. 97, pp. 849-872, 2019. doi: 10.1016/j.future.2019.02.028
[38] R. C. Narayanan, N. Ganesh, R. Čep, P. Jangir, J. S. Chohan and K. Kalita. (2023). A Novel Many-Objective Sine-Cosine Algorithm (MaOSCA) for Engineering Applications. Mathematics, vol. 11, p. 2301, 2023. doi: 10.3390/math11102301
[39] R. Shankar, N. Ganesh, R. Čep, R. C. Narayanan, S. Pal and K. Kalita. (2022). Hybridized particle swarm-gravitational search algorithm for process optimization. Processes, vol. 10, p. 616, 2022. doi: 10.3390/pr10030616
[40] Mr. Nikhil Surkar, Ms. Shriya Timande. (2012). Analysis of Analog to Digital Converter for Biomedical Applications. International Journal of New Practices in Management and Engineering, 1(03), 01 - 07. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/6
[41] W. Wang, C. Ye, P. Yang and Z. Miao. (2020). Research on Movie Recommendation Model Based on LSTM and CNN. 2020. doi: 10.1109/ICCIA49625.2020.00013
[42] A. Handvzic. (2016). Online evaluation of recommender system with MovieLens dataset. JITA-Journal of Information Technology and Aplications, vol. 11, 2016. doi: 10.7251/JIT16020H
[43] M. T. Alam, S. Ubaid, S. S. Sohail, M. Nadeem, S. Hussain, J. Siddiqui and others. (2021). Comparative Analysis of Machine Learning based Filtering Techniques using MovieLens dataset. Procedia Computer Science, vol. 194, p. 210-217, 2021. doi: 10.1016/j.procs.2021.10.075