Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 11, No. 3, August 2020), Pages: 41-59
www.jacr.iausari.ac.ir

Increasing the Accuracy of Recommender Systems Using the Combination of K-Means and Differential Evolution Algorithms

Ali Pazahr✉
Department of Computer Engineering, Ahvaz Branch, Islamic Azad University, Ahvaz, Iran
pazahr@iauahvaz.ac.ir

Received: 2020/07/18; Accepted: 2020/02/10

Abstract

Recommender systems are the systems that try to make recommendations to each user based on performance, personal tastes, user behaviors, and the context that match their personal preferences and help them in the decision-making process. One of the most important subjects regarding these systems is to increase the system accuracy which means how much the recommendations are close to the user interests. In this paper, to achieve the mentioned aim we use a combination of K-means and differential evolution algorithms. The K-means algorithm determines the best recommendations for the current user based on the behavior of the other users. The differential evolution algorithm is used to optimize the user clustering in the recommender system. Given that the proposed model has been tested in a movie domain, the films suggested to the current user, have the highest rates from the users who are similar to the current user. The results gained from the simulation show the superior performance of the proposed model in comparison to the related works with an average increased accuracy of 0.01.

Keywords: *Recommender Systems; K-Means; Differential Evolution Algorithms; Clustering; Accuracy*

1. Introduction

Nowadays, instead of traditional forms, many business activities can be done on the internet and online shop as a kind of e-commerce is one of these activities [1]. Many online shops provide products without any time and space limitations and expose their users to news, films, books, and hundreds of thousands of other products and they choose from them based on their tastes and needs [2]. But when users encounter this vast amount of data, it becomes hard to choose, and searching in this space takes a lot of time from them [3]. The recommender systems are created to solve the vast data problem and help users choose their products [4][5]. But a recommender system is the system that can analyze past behaviors and provide recommendations for new problems [6]. These systems suggest the most appropriate and closest product to the user's taste by guessing the user's way of thinking [7] [8].

One use of the recommender systems is introducing the user's needed resources. These resources could be special information that user needs or products such as books or a user's favorite movie among the infinite information that the user faces [9]. Regarding the explanations, the provided method in this article for increasing the accuracy of the recommender systems is a method based on collaborative filtering [10]. One of the subjects which are the center of attention in the current studies in the field of recommender systems is the combination of the different algorithms to increase the accuracy in these systems [11]. One of the base algorithms which is used for recommender systems is the K-means algorithm [12].

Although several previous research works have tried to use a genetic algorithm with a common accuracy, the evolutionary algorithm used in this paper is the differential evolution algorithm that meets a higher amount of accuracy [13][14][15]. The main difference between the differential evolution and genetics algorithms is in the selection operator [16]. The combination process is such that at first, using the K-means algorithm, various populations, which are incorporated in the clusters of a dataset, are created and then using the differential evolution algorithm and based on the defined fitness function in this algorithm, the optimized population is chosen in the current study.

Although many methods have been suggested for increasing the accuracy of recommender systems, it is still expected that the methods using clustering technics have more accuracy with different number of neighbors [17]. By combining the K-means and differential evolution algorithms, that is the contribution of the current study, the accuracy of the recommender systems is increased and such combination has not been utilized in the previous similar researches [18]. The rest of the paper is organized as follows. In section 2, some similar related works are introduced so that the advances about the accuracy increment of recommender systems are revealed so far. The details about the proposed methodology are highlighted in section 3 describing the components of a recommender system with this method and how they work. In section 4, the proposed method is simulated using simulation in MATLAB and the results of the simulation of this study are mentioned in section 5 comprising the calculation of the evaluation metrics, the comparisons between the suggested method, and other similar works based on the metrics. Finally, in section 6, the conclusion and future works are specified.

2. Related works

One of the researches related to the provided method is to use the genetic algorithm cryptography as a substitute for the K-means algorithm's clustering which targets the reduction of the search space and improving the quality of clustering [19][20][21]. As the results, the proposed method considering precision and recall was compared to two other algorithms including K-means and *k*-NN and both clustering methods had better efficiency than *k*-NN. Although K-means started with better efficiency, the proposed method got better after the neighborhood size of 30. There is a method to recognize the similarities between the users using the evolutionary optimization algorithms based on knowledge called cultural algorithms [22]. By comparing the accuracy of the proposed method with other researches, an outperformance of the suggested method rather than the others was observed by the Pearson Correlation Coefficient, Cosine, and GA algorithms. To suggest movies to users, some methods are provided such as K-means algorithms and K-nearest neighbors, also using the K-means algorithm's clustering and honey bee colony, which are performed similarly [23][24][25]. By using the method of C-MCS, the clusters were more accurate in comparison with the pure K-means [23].

There is a method using the grey wolf algorithm for the recommender systems which rates the movie based on its history data [26][27]. The proposed model had an MAE value of 0.68, less than the other studies. Except for the PCA-K-means algorithm, the proposed method had higher accuracy in comparison to the other 10 algorithms. In the field of differential evolution algorithms, regarding the samples' different penetration levels in cluster analysis, some weights are introduced to design a weighting Euclidean distance [28]. The results proved that the elementary center of the clustering which was calculated by the proposed method has an approximate adoption to the ultimate center and the accuracy of the algorithm was higher than the other compared researches.

In a method, regarding the attributes of the Gaussian distribution, Khushi distribution and the mutation factor are used to improve the searching strategy [29]. The evaluation of the method showed that the accuracy of the CO-JADE method was higher than the HCOEA and COEA/ODE methods as the related works. A multi-part model is suggested, which improves the accuracy and diversity of the recommendations at the same time. By using the global positioning systems' data to produce a trip recommendation, a smart recommender system based on the basic approaches was presented [30]. The experimental results showed that the proposed method had the most Coverage and F-Measure along with the least RMSE in comparison to the other 6 similar works.

The matching of passenger and taxi driver has a critical role in the modern taxi systems. To do so, a two-step matching system of the taxi and passenger is optimized using the differential evolution algorithm to make sure of the service quality and proper profit [31]. Using the proposed model, a reliable, flexible, and efficient schedule algorithm was presented for according taxis and passengers. To make more accurate recommendations, there are methods that use the modified cuckoo search algorithm to optimize the data points in the cluster, the new evolutionary clustering method to gather the users with similar interests in the same cluster, and the K-means clustering algorithm to rate personal products [32]. This investigation showed a value of 0.68 as MAE, less than its previous works. Considering the combination of two algorithms including K-means and PSO, a method proved that this combination can produce the recommendations with higher accuracy in comparison with using only PSO or K-means in the recommendation generation process [33][34][35]. Another method used the combination of K-means and Genetics algorithms [36]. The suggested method was tested on two datasets and the results showed an acceptable accuracy which was observed in other work similarly [37].

As described in this section, different studies have been suggested for increasing the accuracy of recommender systems. However, it is expected not only to decrease the error, but also to increase the accuracy of recommender systems during the increment of the neighbors in the clusters. The proposed method fulfils this issue using a combination of K-means and differential evolution algorithms.

3. Proposed Method

In this work, we propose a novel framework for the recommender systems. In Figure 1, the steps that form part of our recommendation process are depicted. In a nutshell, our approach encompasses two main phases, namely, the clustering phase and the recommendation phase. During the clustering stage, the operations related to clustering are done and the calculations related to the recommendation generations are performed in the recommendation stage.

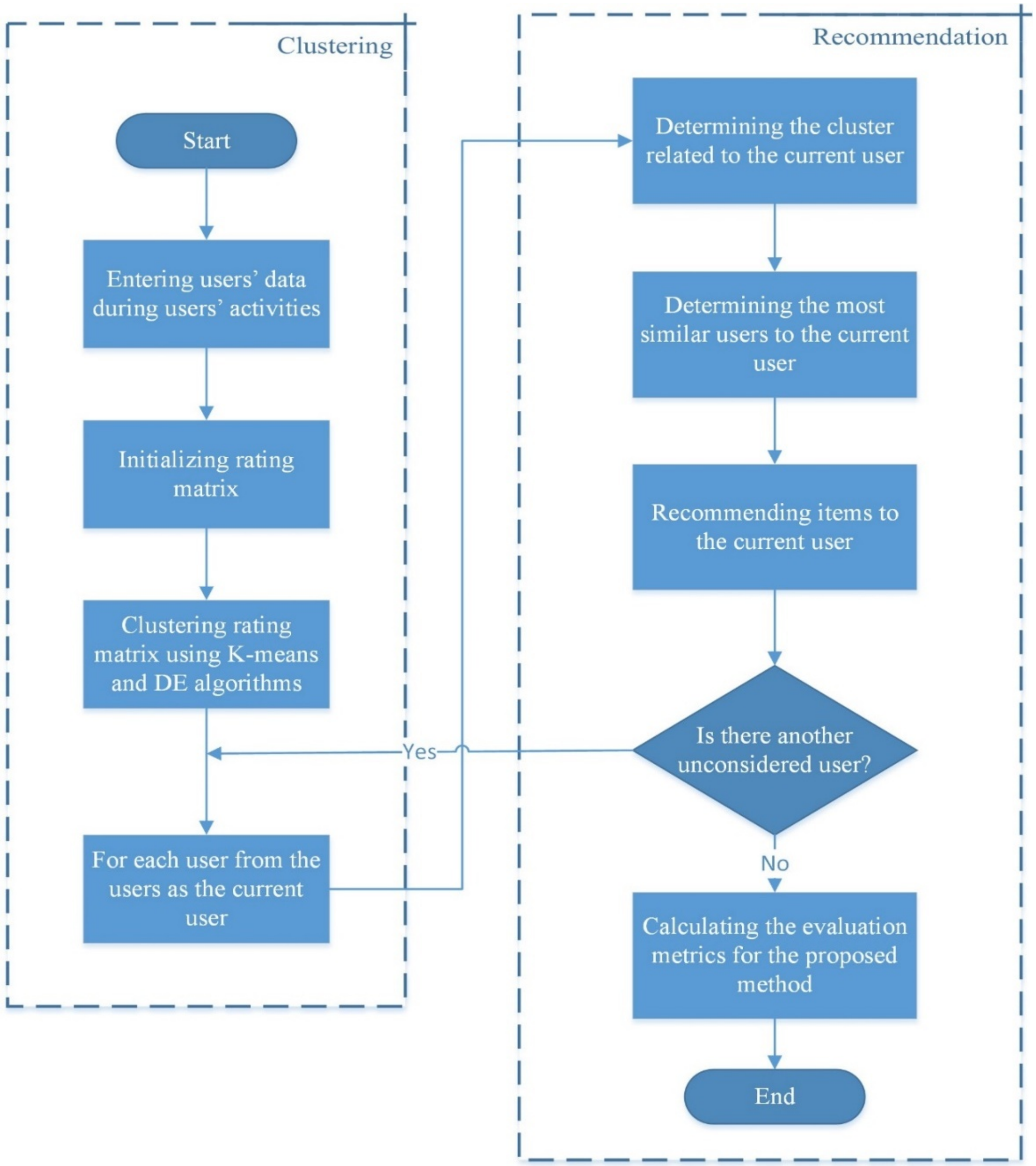

Figure 1: Model of the Proposed method

After presenting the model of the suggested algorithm, in the next parts, more details about the steps of the suggested algorithm are explained.

3.1. Entering users' data:

In this step, the users' opinions regarding different items are entered into the recommender system in the form of a matrix. These items are used to cluster users more precisely. Therefore, similar users based on items are placed in one cluster.

In some categories such as movies, shopping, favorite sport, etc., some ratings are obtained from the user. In fact, in this step, the user will rate the items of a particular field from 1 to 5. These rates show the importance level of the items. This rate setting is created as a matrix based on (1):

$$R(User-Item)=\begin{pmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{m1} & \cdots & d_{mn} \end{pmatrix} \tag{1}$$

Matrix R is known as the user-item matrix or the rating matrix. The variable m is the number of rows, which is the same as the users, and n shows the number of columns, which is the number of the specified items in a particular field. Meanwhile, $d_{i,j}$ represents the rating of the i-th user to the j-th item.

3.2. Initializing the rating matrix:

In this step, the items which the users have rated, are put into an input matrix

3.3. Clustering the rating matrix:

The input rates of the users are entered into the proposed framework. In this step, the K-means algorithm will cluster and produce the initial population. Then, this initial population are given to the differential evolution algorithm, and this algorithm will optimize the clustering algorithm by calculating the fitness of the populations created in several steps.

After creating the rating matrix, user clustering is done using the matrix in Eq. (1). One of the most critical points in clustering algorithms is to determine the appropriate number of clusters. A suitable number of clusters will make the clustering algorithm process more optimized.

In this paper, we will use the silhouette criterion to determine the optimized number of clusters [38]. After determining the optimized number of clusters, the dataset must be clustered using the combination of k-means and differential evolution algorithms. In the k-means algorithm, at first, the initial population is created using (2):

$$X = Uniform\_Random\_Number(Lower, Upper) \tag{2}$$

The Equations (2), (3), and (4) related to simulation, are used in the MATLAB software. In (2), Uniform_Random_Number, which exists in the library functions of MATLAB, will produce a random number with even distribution, X shows the population, lower and upper determine respectively the lower and the upper bound to create regular random numbers.

After initial population creation, as mentioned, the fitness function of the population is calculated using the differential evolution algorithm. The input parameters for this function are generated population (x) and main dataset (y) for clustering. This function is calculated using (3):

$$Distance = Pairwise\_distance\_two\_sets(x, y) \tag{3}$$

The output of the function *Pairwise_distance_two_sets* is the D_{m*n} matrix, which shows the distance between the x and y vectors. After calculating the distance between the two x and y vectors, which is done using the multiply operator, the least distance is chosen as the best answer for this population. Then, amongst the obtained answers from the different states of the algorithm repetition, the least distance is chosen as the best answer, and the vector corresponding to this least answer is considered as the result of clustering. To calculate the new population in the differential evolution algorithm, we must initially define the Mutation and Crossover operators in the suggested algorithm.

The Mutation operator is defined as followed. The Q range is defined as (4):

$$Q = [1 : i-1, i+1 : popsize] \tag{4}$$

In the Eq. (4), *i* shows the loop's counter, and pop size refers to the size of the population. In the next step, the 3 random vectors coming from the *Q* vector must be created. These vectors are defined as (5):

$$Q = [1 : i-1, i+1 : popsize] \tag{5}$$

Then we define A, B, C using (6):

$$A = S(1), B = S(2), C = S(3) \tag{6}$$

After calculating the values based on (6), the Mutation function is defined as (7):

$$U = Population(A) + beta \times (Population(B) - Population(C)) \tag{7}$$

In (7), the population is the created population in the previous step, and the variable beta is defined as (8):

$$beta = Random - Uniform(BetaLB, BetaUB) \tag{8}$$

In (8), Random, from the library function of the MATLAB software, will create a random vector in the range of $BetaLB$ and $BetaUB$. To calculate the Crossover operator, a random number between 1 and $nvar$ is created and $nvar$ is the multiplication result of NE[1] and NC[2] which respectively are the number of entries and the number of clusters. As followed, a random set including 1 to $nvar$ and (9) is evaluated:

$$Random - Set(1, n\text{var}) < Pcross \tag{9}$$

The Eq. (9) produces the location of the numbers, which are less than a crossover probability from the random numbers created in the range between 0 and 1. Consequently, based on the (10), the new population is created:

$$\begin{aligned} &Index = [Random - number, Random - set]; \\ &x(Index) = U(Index) \end{aligned} \tag{10}$$

In (10), U is the set obtained from the Mutation operator.

3.4. Determining the cluster related to the current user

In this step, based on the gathered information from the previous section about the clusters, the cluster corresponding to the current user is specified.

3.5. Indicating the most similar users to the current user

After clustering users in the clusters, in this step, the ratings of the current user must be approximated. To do so, first, we must determine the cluster in which the user is. After specifying the cluster, the k users who are the most similar to the current user are identified, and then, based on the average of their ratings, the rating of the current user is approximated. The process of calculating the similarity criteria between two users in the proposed model is accomplished using Euclidean distance in (11):

$$D_r(a, b) = \sqrt{\sum_{j=1}^{r} (a_i - b_i)^2} \tag{11}$$

[1] Number of Entries
[2] Number of clusters

a_i and b_i show similar features for two users, and D determines the size of the problem aspects, and r refers to the aspects which are affecting the calculation of the similarities [39]. Furthermore, for finding the most similar users to the current user, it is possible to use the k-NN algorithm.

3.6. Recommending items to the current user

After that, the current user is put in his cluster, and the most similar neighbors are found, the items are sorted based on the similarity and the level of ratings. Then, based on the number of suggested items to similar users, the items are recommended to the current user.

3.7. Evaluating the suggested method:

After the ratings of the current users to the recommended items are calculated using the k-nearest neighbor algorithm, the evaluation metrics for the suggested algorithm should be calculated, i.e. accuracy, recall, the error level.

4. Simulation

The simulation phase of the model consists of preparing a system for running the methodology based on a dataset and observing how it works.

4.1. System specifications

All simulations and case studies have been done on a system with a 2.53 GHz processor, 4 GB of RAM, and Windows 10 operating system. MATLAB programming language and simulation environment have been used to simulate the suggested algorithm.

4.2. Dataset description

This dataset is taken from the MovieLens[3]. The file contains 1000209 records including 943 users and 1682 movies, which users rated in 2000. All the ratings are in ratings.dat file that is defined as the following structure:

UserID: MovieID: Rating: Timestamp

Information of the users exists in the Users.dat file, which is shown as below:

UserID: Gender: Age: Occupation: Zip-code

The age range of the users is according to Table 1:

Table 1: The age ranges of the users

Age range	Shown number	Age range	Shown number
Under 18	1	18 - 24	18
25 - 34	25	35 - 44	35
45 - 49	45	50 - 55	50
Older than 56	56		

The *Occupation* parameter indicates the occupation of the user, which is specified as Table 2:

[3] https://grouplens.org/datasets/movielens

Table 2: The occupation of the users

Occupation status	Indicating number	Occupation status	Indicating number
other	0	educator	1
Artist	2	admin	3
college	4	Customer service	5
doctor	6	managerial	7
farmer	8	homemaker	9
K-12 student	10	lawyer	11
programmer	12	retired	13
sales	14	scientist	15
self-employed	16	technician	17
Tradesman	18	unemployed	19
writer	20		

The information of the movies exists in the Movies.dat file, and each movies' information is shown in the format below:

MovieID::Title::Genres

Regarding the explanation given for the dataset, in this paper, a dataset including 943 users who have seen 1682 movies and rated them is used. Considering the given ratings to the movies by different users, a rating matrix consisting of n user and m movies is created, which each cell of this matrix is the rating given by the user to that particular movie. This matrix is created using the rating that users give to movies in the MovieLens website to different movies.

4.3. Simulating the suggested algorithm:

In the first step of simulating the suggested algorithm, the best number of clusters must be determined using the silhouette value [40]. The calculation of silhouette value is performed on the selected dataset. To do so, we start from value 2 and repeat it for the other cluster numbers until 10, and finally, we compare the obtained silhouette values. Accordingly, the Z, which gives the highest value of silhouette, is chosen as the best value. Table 3, shows the number of Z and the value of silhouette for each Z:

Table 3: The values of the silhouette for Z

Z value	Silhouette value	Z value	Silhouette value
2	0.2357	3	0.2505
4	0.2104	5	0.0514
6	0.1198	7	0.0455
8	0.0201	9	0.0349
10	0.0531		

Considering Table 3, for Z=3 we have the highest value of silhouette, and therefore Z=3 is chosen as the best value for clustering the dataset. In the next step of simulation, we apply the suggested algorithm on the dataset and analyze the results. Table 4 will specify the input values for the suggested algorithm.

Table 4: Input values for the suggested algorithm

variable	value

number of clusters	3
Number of users	943
Number of movies	1682
The number of neighbors	4 to 64
Number of recommendations	20
Number of repetitions	100
Population size	10

The dataset is categorized into 3 clusters. The number of users is 943, and the number of movies is 1682. When the current user logs in the system and gets categorized in a cluster, using the K-nearest neighbor algorithm, we determine the best neighbors for the current user, and based on these users, movies are suggested to the user. The number of recommendations to the current user is 20. The size of the population in the differential evolution clustering algorithm is 10, and the number of repetitions in this method is 100.

5. Evaluation and Results

After executing the clustering application, which is specified in Table 4, the clustering is done, and the Best value is shown in Figure 2.

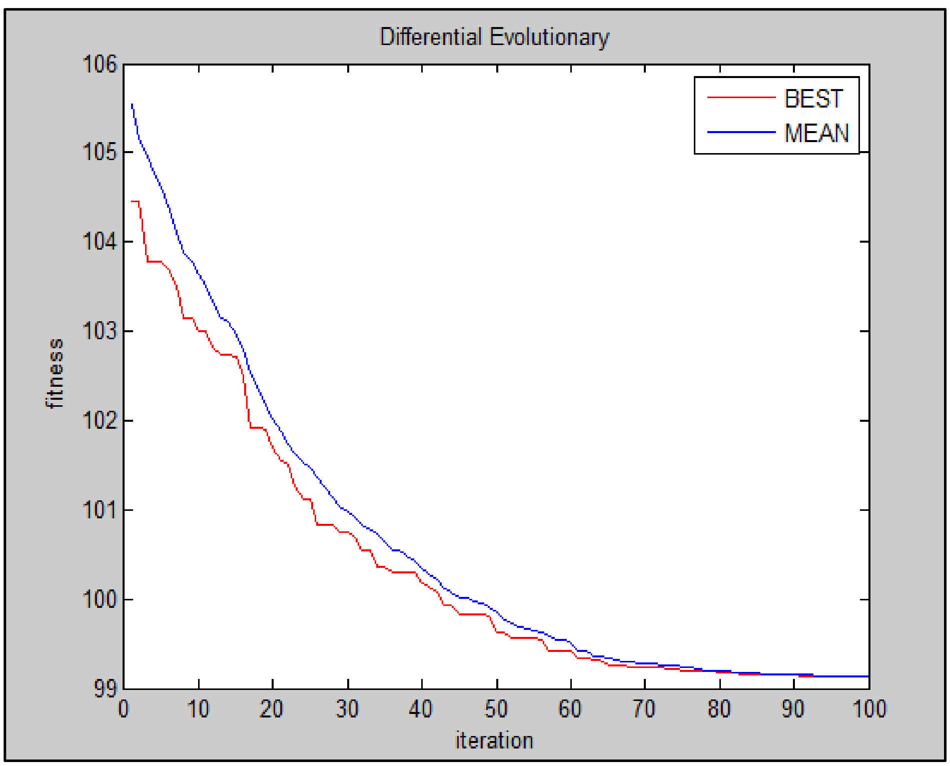

Figure 2: The value of Best for each repetition of the suggested algorithm

Regarding Figure 2, the Best value in each repetition is shown. This value shows the inner cluster distance, which is reduced in each repetition, and this reduction process is a favorable result for the suggested algorithm. The Mean vector shows the middle of distances in each cluster after each repetition of the algorithm. After finishing the execution of the suggested algorithm, for example, for user number 3, the recommendations are as Table 5:

Table 5: The recommendations for user number 3

Column 1 to 12												
	1	2	4	6	7	8	9	10	11	12	13	14
Column 13 to 20												
					15	16	17	19	22	23	24	25

These recommended movies have the highest ratings from the users most similar to user number 3. The reason for most of these ratings is that the most similar neighbors to the current user are specified. Then, the recommendations based on the movies that neighbors have watched and rated the highest, are made to the current user. After executing the suggested algorithm, in this section, the evaluation metrics such as Accuracy (14), error, the RMSE (13), and MAE (12) should be calculated and analyze the results. To evaluate the suggested method, these metrics are used:

MAE shows the average difference between the real value and predicted value which is explained by Eq. 12:

$$MAE = \frac{\sum_{i=1}^{n} \left| value_{actual} - value_{predicted} \right|^2}{n} \tag{12}$$

Value$_{actual}$ shows the real rating of the user to the item and value$_{predicted}$ shows the suggested rating by the recommender system.

RMSE shows the average difference between the real value and predicted value and is explained by (13):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (value_{actual} - value_{predicted})^2}{n}} \tag{13}$$

Accuracy is one of the most important factors in decision-making systems algorithms which are calculated as (14):

$$Accuracy = 1 - \left(\frac{\sum_{i=1}^{n} \left| value_{actual} - value_{predicted} \right|}{n}\right) \tag{14}$$

First, the error average of the recommendation made by the recommender system is calculated, and then this value is subtracted from the value 1, which gives us the accuracy measure. Table 6, shows the results of the evaluation:

Table 6: The results of the evaluation

Metric	Value
Accuracy	0.73943
Error	0.26057
RMSE	1.1473
MAE	1.5289

In this paper, using the holdout categorization method, a part of records (50%) is considered for the training and the other part (50%) to test the suggested model [41]. Therefore, to explain the outputs of the program using the test records of a running program is shown in Figure 3 which are divided into 9 groups of 50:

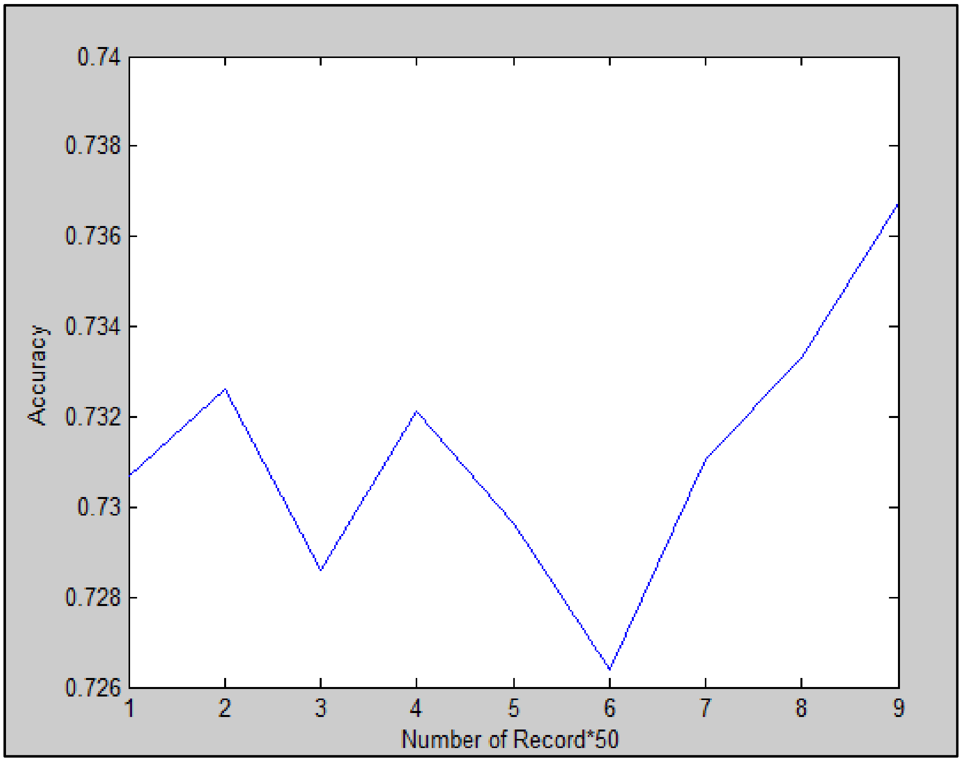

Figure 3: Program outputs using the test records

Moreover, as observable in Figure 3, as the records are different in each category, from category 6, the Accuracy has increased. In categories 1 and 3, the Accuracy of recommendations to users has increased, and this value is reduced in categories 2,4,5. The reason for this increase and decrease could be similar neighbors to the new user and the movies that these neighbors have watched and are suggested to the current user. The graph in Figure 4, shows the error metric for the different numbers of records. This criterion is calculated by subtracting the Accuracy from 1. As an example, for accuracy of 0.731, the error value is 1-0.731=0.269.

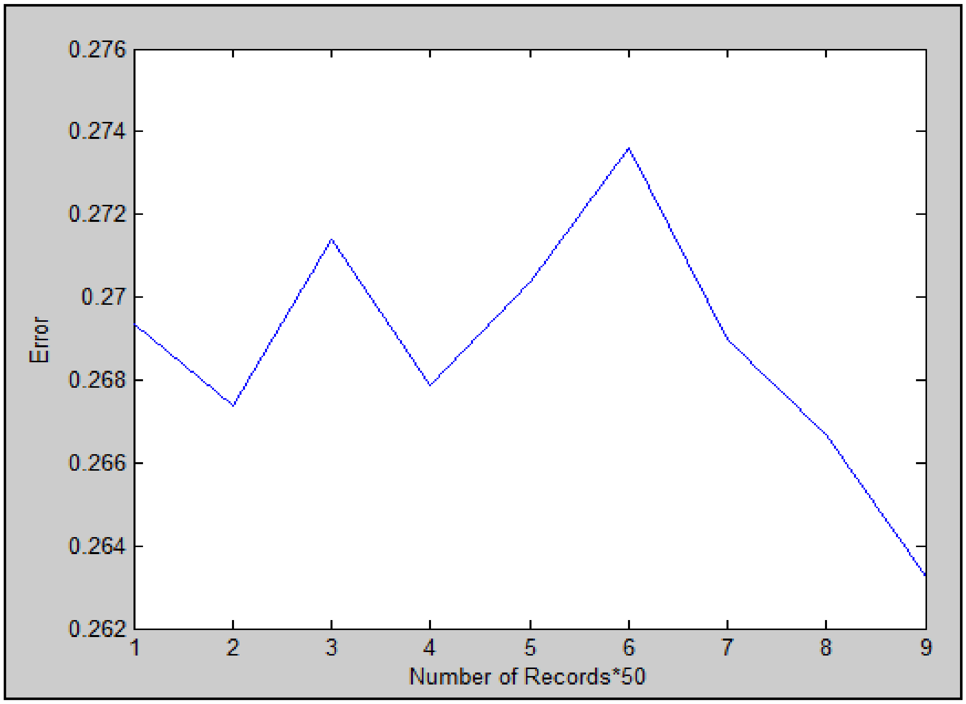

Figure 4. The error graph for different number of test records

As seen in Figure 4, by increasing the records of the dataset, the error value is increased at first and then decreased, which shows that the recommendations error to new users in the system is increased at first and then decreased by having the correctly suggested movies. The horizontal axis shows the number of records in groups of 50. Figure 5 shows the MAE.

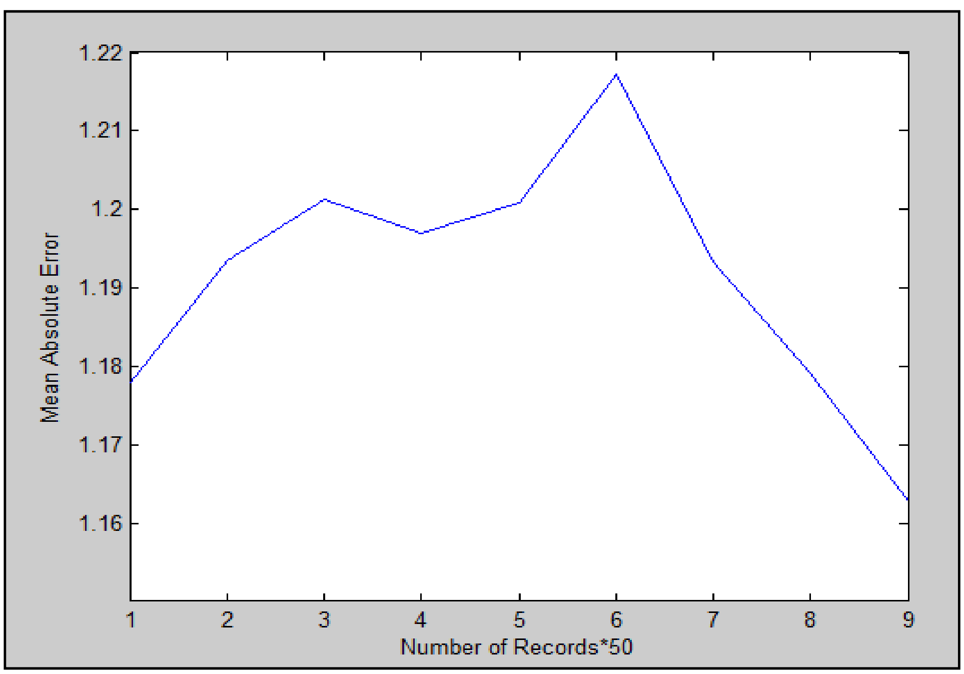

Figure 5. MAE for different repetition of the records of the test dataset

As seen in Figure 5, by increasing the number of records in the test dataset, the factor of MAE of the suggested program is increased at first and then decreased which means that the movies suggested to the users of the primitive categories have an increasing error rate, and film suggested to the latter categories are more accurate, and this has led to error reduction. The graph in Figure 6, correctly shows the RMSE.

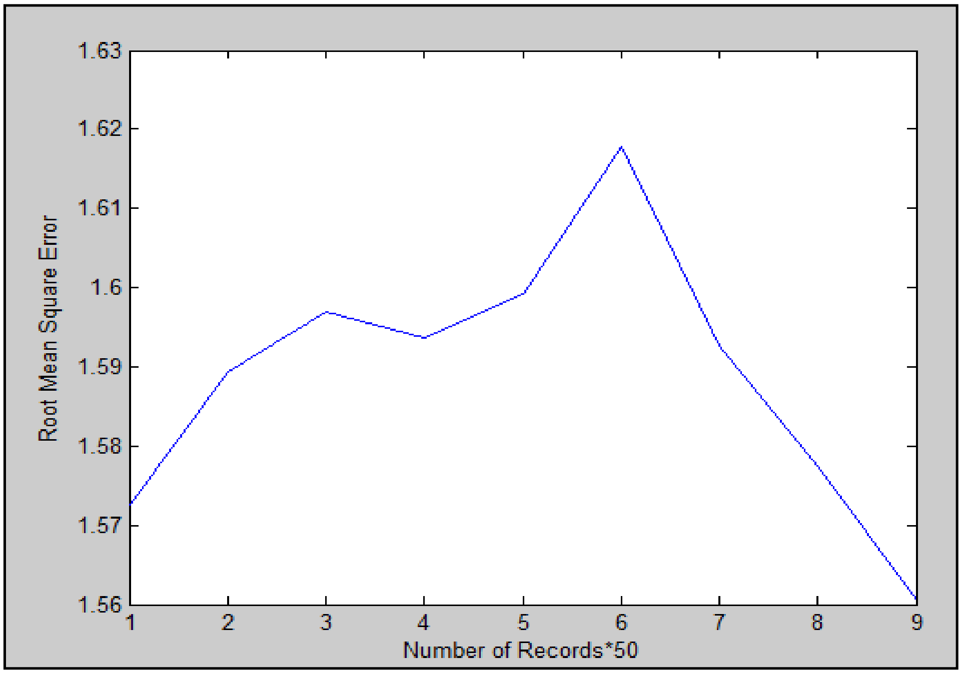

Figure 6. The RMSE for different repetition of the records of the test dataset

As seen in the graph in Figure 6, the RMSE at first increased by adding the records of the test dataset and then reduced. The reason is that the moves suggested to the users of the first categories have more error levels, and movies suggested to the late categories have less error level. After analyzing the results of simulating the suggested algorithm, in this step, the results were compared to the "K-means and PSO" and "K-means and Genetics" on the dataset. Table 7 shows the comparison:

Table 7: Comparing the suggested algorithm to the other algorithms based on MAE

Factor	Number of neighbors	K-means and PSO	K-means and Genetics	Proposed method
MAE	4	0.91022	0.90795	0.906
	8	0.85972	0.85665	0.85481
	16	0.83373	0.82467	0.81573
	32	0.8069	0.81285	0.8022
	64	0.79725	0.8121	0.79453

The graph in Figure 7 shows this comparison in a better way.

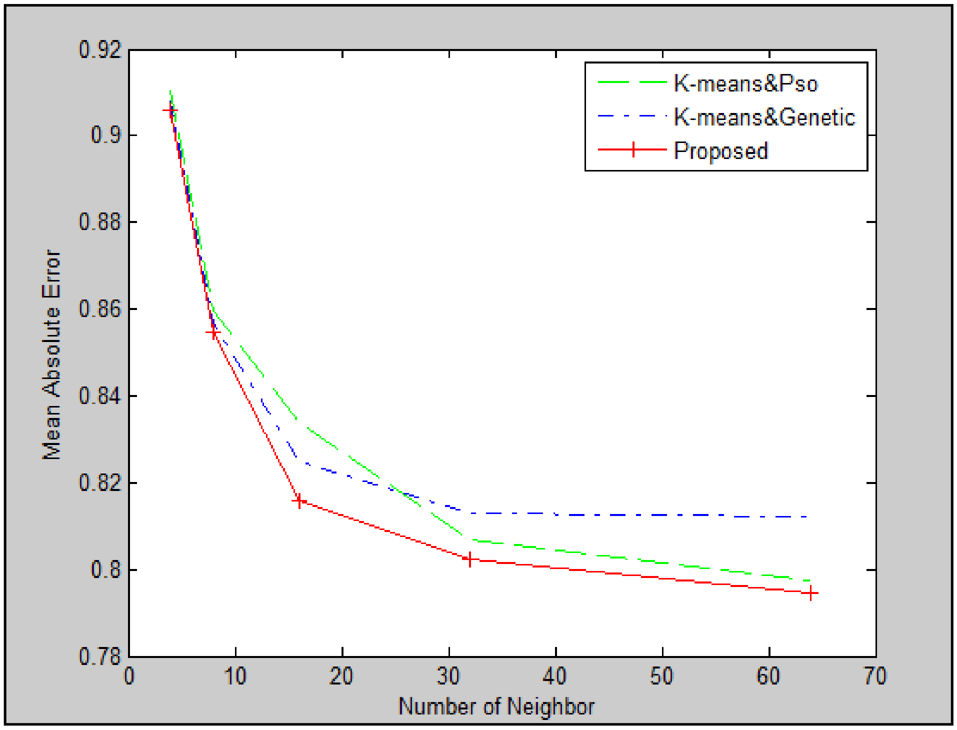

Figure 7. The comparison of MAE factor for different methods

As seen in Figure 7, the MAE for the different number of neighbors is less than and more optimized than the other methods presented in Table 7.

Table 8: Comparing the suggested algorithm to the other algorithms based on the RMSE

Factor	Number of neighbors	K-means and PSO	K-means and Genetics	Proposed method
RMSE	4	1.1839	1.1838	1.1822
	8	1.13	1.1363	1.1354
	16	1.1207	1.12	1.1197
	32	1.0998	1.1061	1.0599
	64	1.0868	1.0996	1.0749

Figure 8, shows the comparison of the RMSE for suggested and other methods based on the results of Table 8:

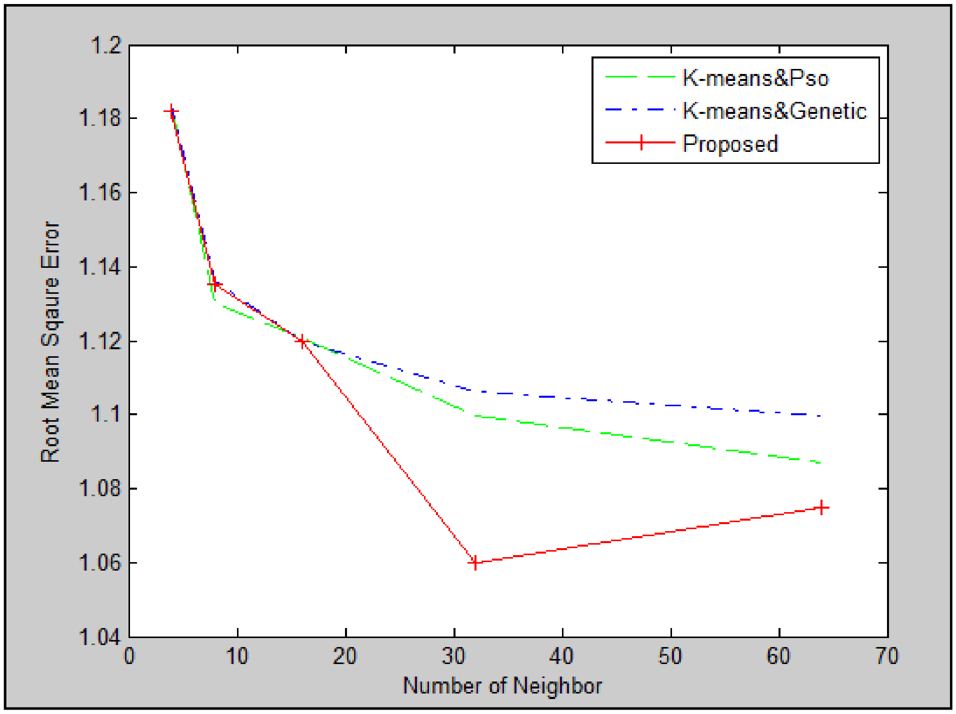

Figure 8. Comparison of the RMSE for the proposed and similar methods

As seen in the graph in Figure 8, the RMSE for the suggested method is less and more optimized than the other methods. A comparison of the accuracy of the proposed method to the other researches based on the number of neighbors has been shown in Table 9.

Table 9: Comparing the accuracy of the suggested algorithm to the other methods

Factor	Number of neighbors	K-means and PSO	K-means and Genetics	Proposed method
Accuracy (Neighbor)	4	0.2012	0.2013	0.2141
	8	0.24945	0.2497	0.2601
	16	0.31355	0.3099	0.31628
	32	0.35235	0.34345	0.3664
	64	0.35935	0.3435	0.36425

Figure 9, clearly shows the comparison graph of the suggested method of other methods:

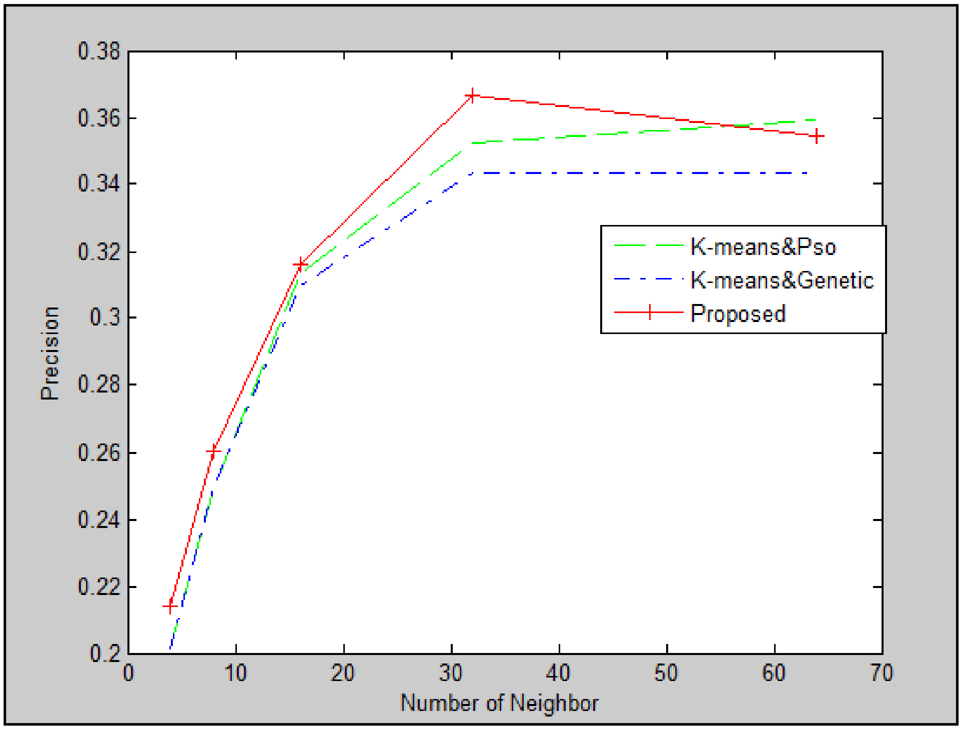

Figure 9. Comparison of accuracy for the suggested method and the other methods

As seen in Figure 9, the accuracy factor for the different number of neighbors is more and more optimized than the other methods.

6. Conclusion and Future works

In this paper, a method for increasing the accuracy of the recommender systems was presented using a combination of K-means and differential evolution algorithms. The K-means algorithm specified the most suitable recommendations for the current user considering the other users' interactions. For optimizing the user clustering, the differential evolution algorithm was utilized. The inner cluster distance was reduced in each repetition, and this reduction process was a favorable result for the suggested algorithm. Using the differential evolution algorithm in the recommender systems, the K-means clustering algorithm was improved. The current model was compared with two similar methods which were the algorithms using "K-means and PSO" and "K-means and Genetics" in terms of three evaluation metrics including MAE, RMSE, and Accuracy. The results showed that not only the accuracy of recommendations in the proposed model was increased in each category of neighbors using the combination of the differential evolution and k-means algorithms, but also the MAE and RMSE were decreased in comparison with the similar methods. Briefly, an average increased accuracy of the proposed method was 0.01.

The current research proposes an algorithm to increase the accuracy of the generated recommendations so that the users can be more satisfied with the recommendations because such recommendations can be closer to the users' preferences using the proposed method. As a result, the applications which use these recommender systems can fascinate the users more efficiently and it can be led to more attraction of audiences and clients.

Therefore, the related businesses using the suggested algorithm can be more successful in e-commerce and e-marketing with more sales. Such recommendations can provide value-added services along with more sales diversity and competitive pricing. Furthermore, the advertisement can be more accurate and targeted.

Since the biggest challenge of using the recommender systems is that these systems are not currently able to give the best recommendations with the highest accuracy, by considering the mentioned description about the suggested method and stating the pros and cons of these methods, some suggestions for further future researches could be made:

- Using other evolutionary algorithms such as grey wolf or dragonfly to replace the differential evolution algorithm for increasing the accuracy and speed of clustering. Each one of these algorithms has advantages that could be used to improve the suggested system.
- Using a neural network to reply faster to the current user based on analyzing the ratings of the previous users in the system to replace with the k-nearest neighbor algorithm. The neural network can be used to increase the response speed of the recommendation generation process for the users. The neural network can increase the speed of the recommender system by training via the existing data.
- Using other similarity matrix measurements to increase the accuracy of the recommender systems.
- To determine the similarity level of the current user to the previous users in the clusters to increase the accuracy of clustering and accordingly the accuracy of the generated recommendations, some similarity calculation formulas via different ways could be provided.

References

[1] R. Ahuja, A. Solanki, and A. Nayyar, "Movie recommender system using k-means clustering and k-nearest neighbor," *Proc. 9th Int. Conf. Cloud Comput. Data Sci. Eng. Conflu. 2019*, pp. 263–268, 2019.

[2] A. Marchand and P. Marx, "Automated Product Recommendations with Preference-Based Explanations," *J. Retail.*, 2020.

[3] D. Wu, M. Shang, G. Wang, and L. Li, "A self-training semi-supervised classification algorithm based on density peaks of data and differential evolution," *ICNSC 2018 - 15th IEEE Int. Conf. Networking, Sens. Control*, pp. 1–6, 2018.

[4] C. E. Berbague, N. E. I. Karabadji, and H. Seridi, *Enhancing the Sales Diversity Using a Two-Stage Improved KNN Algorithm*, vol. 64. Springer International Publishing, 2019.

[5] F. Soleymani Sabzchi, S. Jamali, and M. Jafari, "Mining users ` navigation patterns for building web pages recommendation system," *J. Adv. Comput. Res.*, vol. 4, no. 2, pp. 15–24, 2013.

[6] K. Kanagarajan and S. Arumugam, "Intelligent sentence retrieval using semantic word based answer generation algorithm with cuckoo search optimization," *Cluster Comput.*, vol. 22, pp. 7003–7013, 2019.

[7] M. Kiani Sarkaleh and F. Khoshnood, "Designing a Trust-Based Recommender System in Social Rating Networks," *J. Adv. Comput. Res.*, vol. 10, no. 2, pp. 75–89, 2019.

[8] M. Bałchanowski, U. Boryczka, and K. Dworak, "Using Differential Evolution with a Simple Hybrid Feature for Personalized Recommendation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10751 LNAI, pp. 137–146, 2018.

[9] J. Li *et al.*, "Category Preferred Canopy–K-means based Collaborative Filtering algorithm," *Futur. Gener. Comput. Syst.*, vol. 93, pp. 1046–1054, 2019.

[10] F. Shomalnasab, M. Sadeghzadeh, and M. Esmaeilpour, "An Optimal Similarity Measure for Collaborative Filtering Using Firefly Algorithm," *J. Adv. Comput. Res.*, vol. 5, no. 3, pp. 101–111, 2014.

[11] N. Tohidi and C. Dadkhah, "Improving the performance of video Collaborative Filtering Recommender Systems using Optimization Algorithm," *Int. J. Nonlinear Anal. Appl.*, vol. 11, no. 1, pp. 283–295, 2020.

[12] H. T. Cheng *et al.*, "Wide & deep learning for recommender systems," *ACM Int. Conf. Proceeding Ser.*, vol. 15-Septemb, pp. 7–10, 2016.

[13] R. Kaluri and C. P. Reddy, "A framework for sign gesture recognition using improved genetic algorithm and adaptive filter," *Cogent Eng.*, vol. 3, no. 1, p. 1251730, 2016.

[14] B. S. Neysiani, N. Soltani, R. Mofidi, and M. H. Nadimi-Shahraki, "Improve performance of association rule-based collaborative filtering recommendation systems using genetic algorithm," *Int. J. Inf Technol. Comput. Sci.*, vol. 2, pp. 48–55, 2019.

[15] H. Parvin, P. Moradi, and S. Esmaeili, "A collaborative filtering method based on genetic algorithm and trust statements," in *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, 2018, pp. 13–16.

[16] P. Lou, G. Zhao, X. Qian, H. Wang, and X. Hou, "Schedule a rich sentimental travel via sentimental poi mining and recommendation," *Proc. - 2016 IEEE 2nd Int. Conf. Multimed. Big Data, BigMM 2016*, pp. 33–40, 2016.

[17] P. K. R. Maddikunta, T. R. Gadekallu, R. Kaluri, G. Srivastava, R. M. Parizi, and M. S. Khan, "Green communication in IoT networks using a hybrid optimization algorithm," *Comput. Commun.*, 2020.

[18] Z. Yang, B. Wu, K. Zheng, X. Wang, and L. Lei, "A survey of collaborative filtering-based recommender systems for mobile internet applications," *IEEE Access*, vol. 4, no. c, pp. 3273–3287, 2016.

[19] M. Tahir, M. Sardaraz, Z. Mehmood, and S. Muhammad, "CryptoGA: a cryptosystem based on genetic algorithm for cloud data security," *Cluster Comput.*, pp. 1–14, 2020.

[20] C. Berbague, N. Karabadji, and H. Seridi-Bouchelaghem, "An Evolutionary Scheme for Improving Recommender System Using Clustering," 2018, pp. 290–301.

[21] A. Pazahr, J. J. Samper-Zapater, F. García-Sánchez, C. Botella, and R. Martinez, "Semantically-enhanced advertisement recommender systems in social networks," in *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services*, 2016, pp. 179–189.

[22] P. Kaufmann, P. A. C. Eds, I. Conference, and D. Hutchison, *A Cultural Algorithm for Determining Similarity Values Between Users in Recommender Systems*, vol. 11454. Springer International Publishing, 2019.

[23] S. P. Singh and S. Solanki, "A Movie Recommender System Using Modified Cuckoo Search," in *Lecture Notes in Electrical Engineering*, 2019, vol. 545, pp. 471–482.

[24] V. Subramaniyaswamy, R. Logesh, M. Chandrashekhar, A. Challa, and V. Vijayakumar, "A personalised movie recommendation system based on collaborative filtering," *Int. J. High Perform. Comput. Netw.*, vol. 10, no. 1–2, pp. 54–63, 2017.

[25] R. Katarya, "Movie recommender system with metaheuristic artificial bee," *Neural Comput. Appl.*, vol. 30, no. 6, pp. 1983–1990, 2018.

[26] R. Katarya and O. P. Verma, "Recommender system with grey wolf optimizer and FCM," *Neural Comput. Appl.*, vol. 30, no. 5, pp. 1679–1687, 2018.

[27] M. Kumar, D. K. Yadav, A. Singh, and V. Kr., "A Movie Recommender System: MOVREC," *Int. J. Comput. Appl.*, vol. 124, no. 3, pp. 7–11, 2015.

[28] F. Wang, "A Weighted K-means Algorithm Based on Differential Evolution," *Proc. 2018 2nd IEEE Adv. Inf. Manag. Commun. Electron. Autom. Control Conf. IMCEC 2018*, no. Imcec, pp. 2269–2274, 2018.

[29] K. Li, L. Zhong, L. Zuo, and Z. Wang, "Constrained evolution algorithm based on adaptive differential evolution," *Int. J. High Perform. Comput. Netw.*, vol. 11, no. 3, pp. 223–230, 2018.

[30] R. Logesh, V. Subramaniyaswamy, and V. Vijayakumar, "A personalised travel recommender system utilising social network profile and accurate GPS data," *Electron. Gov.*, vol. 14, no. 1, pp. 90–113, 2018.

[31] Y.-J. Gong, Y.-W. Liu, Y. Lin, W.-N. Chen, and J. Zhang, "Real-Time Taxi-Passenger Matching Using a Differential Evolutionary Fuzzy Controller," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. PP, pp. 1–14, 2019.

[32] R. Katarya and O. P. Verma, "An effective collaborative movie recommender system with cuckoo search," *Egypt. Informatics J.*, vol. 18, no. 2, pp. 105–112, 2017.

[33] P. Santos *et al.*, "Application of PSO-based clustering algorithms on educational databases," *2017 IEEE Lat. Am. Conf. Comput. Intell. LA-CCI 2017 - Proc.*, vol. 2017-Novem, pp. 1–6, 2018.

[34] E. Shafiei Gol, A. Ahmadi, and A. Mohebi, "Intelligent Approach for Attracting Churning Customers in Banking Industry Based on Collaborative Filtering," *J. Industerial Syst. Eng.*, vol. 9, no. 4, pp. 9–25, 2016.

[35] B. Uzzi and J. Spiro, "Collaboration and creativity: The small world problem," *Am. J. Sociol.*, vol. 111, no. 2, pp. 447–504, 2005.

[36] V. Soundarya, "Recommendation System for Criminal Behavioral Analysis on Social Network using Genetic Weighted K-Means Clustering," *J. Comput.*, vol. 12, no. 3, pp. 212–220, 2017.

[37] K. jae Kim and H. Ahn, "A recommender system using GA K-means clustering in an online shopping market," *Expert Syst. Appl.*, vol. 34, no. 2, pp. 1200–1209, 2008.

[38] X. Li, W. Liang, X. Zhang, S. Qing, and P.-C. Chang, "A cluster validity evaluation method for dynamically determining the near-optimal number of clusters," *Soft Comput.*, vol. 24, no. 12, pp. 9227–9241, 2020.

[39] R. Kaluri and P. R. CH, "Optimized feature extraction for precise sign gesture recognition using self-improved genetic algorithm," *Int. J. Eng. Technol. Innov.*, vol. 8, no. 1, pp. 25–37, 2018.

[40] D.-T. Dinh, T. Fujinami, and V.-N. Huynh, "Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient," in *International Symposium on Knowledge and Systems Sciences*, 2019, pp. 1–17.

[41] W. Tansey, V. Veitch, H. Zhang, R. Rabadan, and D. M. Blei, "The holdout randomization test: Principled and easy black box feature selection," *arXiv Prepr. arXiv1811.00645*, 2018.