Temporal-based Optimization to Solve Data Sparsity in Collaborative Filtering

Ismail Ahmed Al-Qasem Al-Hadi[1]*, Mohammad Ahmed Alomari[2]
Eissa M. Alshari[3], Waheed Ali H. M. Ghanem[4], Safwan M Ghaleb[5]
Faculty of Ocean Engineering Technology and Informatics Universiti Malaysia Terengganu, Terengganu, Malaysia[1, 4, 5]
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia[2]
Department of Data Science, Universiti Malaysia Kelantan, Kota Bharu 16100, Kelantan, Malaysia[3]

Abstract—**Collaborative Filtering (CF) is a widely used technique in recommendation systems. It provides personal recommendations for users based on their preferences. However, this technique suffers from the sparsity issue which occurs due to a high proportion of missing rating scores in a rating matrix. Several factorization approaches have been used to address the sparsity issue. Such techniques have also been considered to tackle other challenges such as the overfitted predicted scores. Nevertheless, they suffer from setbacks such as drift in user preferences and items' popularity decay. These challenges can be solved by prediction approaches that accurately learn the long-term and short-term preferences integrated with factorization features. Nonetheless, the current temporal-based factorization approaches do not accurately learn the convergence of the assigned k clusters due to a lower number of short-term periods. Additionally, the use of optimization algorithms in the learning process to reduce prediction errors is time-consuming which necessitates a faster optimization algorithm. To address these issues, a new temporal-based approach named TWOCF is proposed in this paper. TWOCF utilizes the elbow clustering method to define the optimal number of clusters for the temporal activities of both users and items. This approach deploys the whale optimization algorithm to accurately learn short-term preferences within other factorization and temporal features. Experimental results indicate that TWOCF exhibits a superior CF prediction accuracy achieved within a shorter execution time when compared to the benchmark approaches.**

Keywords—Collaborative filtering; matrix factorization; temporal-based approaches; whale optimization

I. Introduction

Nowadays, recommendation systems have become popular as they efficiently suggest items to customers according to their feedback and interests [1]. The major resources (data) used to create recommendations are customer profiles, item profiles, and user-item connections (i.e., customer scores to the suggested items) [2]. Collaborative filtering (CF), content-based filtering, demographic filtering, and hybrid filtering are four forms of filters employed in recommendation systems [3]. CF is one of the best prevalent recommendation techniques that provide users with personalized predictions based on their preferences. It relies solely on past users' rating scores on products and does not require the creation of explicit profiles. For example, CF utilizes the rating scores of neighbours to predict a list of items to the active user. However, CF suffers from three main issues: data sparsity [3], [4], cold start [5], [6], and scalability [7] [8]. This paper briefly discusses the approaches utilized to solve the sparsity issue.

Normally, CF-based recommendation systems arrange customers' product rating scores in form of a rating matrix. Customers rank a small number of products using rating scores. These scores are then arranged into a rating matrix. The rating matrix contains very few scores while others are unknown or sparse. This reduces the prediction accuracy of the CF technique. CF provides a list of recommendations to an active user based on his/her interests and according to the feedback of common users who rate some items that are rated by the active user. The feedback is calculated using the similarity and prediction assessments. The similarity assessment between common users and the active user will be infeasible or inconsistent if there is a higher proportion of missing rating scores in the rating matrix [9].

The optimization algorithms have proven successful in several areas such as healthcare [10], document processing [11], and recommendation systems [12]. Various factorization approaches have been used to solve the sparsity issue. These include an imputation-based matrix factorization [2], ensemble divide and conquer [13], and neighbourhood matrix factorization [14]. Although these approaches can learn factorization and latent features that influence the prediction accuracy of the CF technique [12], they cannot effectively learn temporal behaviors and temporal issues such as the drift in users' preferences and the popularity decay of items [15] [16].

The temporal collaboration model [17] merged factorization vectors, long-term preferences, and short-term preferences to enhance the efficiency of the CF technique. The short-term feedback is defined using the shrunk neighbors approach [18] [19] [17]. Furthermore, the short-term preferences are defined by a timestamp factor that determines time periods such as the number of years, seasons, months, and so on. This duration is used to assign *k* clusters to learn the short-term features using *k*-means clustering algorithm. The temporal-based approach [1] achieves higher prediction performance compared to other previous temporal approaches. However, this approach cannot be implemented on the small rating matrix since the *k*-clustering cannot be achieved (*k* problem).

*Corresponding Author

Also, bacteria foraging optimization algorithm has been used to learn the accurate temporal and factorization features [16], [15]. This algorithm provides different error values (fitness) in the search space. The error value increases in some iterations while in others, it decreases; thus, consuming time. To address all these challenges, a new temporal-based whale optimization approach named TWOCF is proposed in this paper.

The TWOCF approach uses an elbow clustering method [20] to accurately learn the precise number of clusters in the time matrix of users' activities. Additionally, TWOCF approach learns the accurate weights for short temporal features that are integrated with other factorization and long temporal features.

The unique aspects of this research can be summarized as follows:

- Solving the sparsity problem.
- Learning accurate latent features throughout the learning iterations.
- Solving the drift and decay issues.
- Reducing the running time of the learning process.

The factorization-based optimization approaches [15] [16] [1] have been focused on solving the temporal and sparsity issues based on personalization. These approaches have improved the accuracy of predicted rating scores while ignored the running time. Despite running time is a significant factor in online recommendation systems, most of the recommendation-based optimization approaches ignore running time. Therefore, TWOCF approach is proposed to solve temporal and sparsity issues with the purpose of improving the prediction accuracy and reducing running time.

The rest of the paper is constructed thus: Section 2 discusses the related works (including the factorization and temporal methods) and describes the whale optimization algorithm (WOA) whose methodology is explained in Section 3. In Section 4, the investigational findings are explained while Section 5 summarizes the main findings of this study.

II. Related Works

A. Matrix Factorization

Matrix factorization has been recently used for solving the sparsity and cold start issues associated with the CF technique in recommendation systems [21]. Matrix factorization is characterized by two features: baseline and latent features. The latent features are defined using the singular value decomposition algorithm [22]. Many factorization methods integrate latent and baseline features of users and items utilizing several formulae [7]. For example, the baseline formula can be used to predict the missing rating scores in the rating matrix as shown in Equation (1).

$$\widehat{r}_{uv} = \mu + B_u + B_v + p_u q_v^{\tau}, \tag{1}$$

where $\widehat{r}_{uv}$ is the expected value for the sparse score, μ is the global rate of all rating scores, B_u and B_v are the base values of users and items, respectively. The factors p_u and q_v^{τ} are the latent feedback of users and items, respectively. Additionally, the norm latent factor is used in several methods [23]. Equation (2) is another example where various latent features are integrated with the weight of γ to minimize the overfitting in the predicted rating scores.

$$\min \sum_{(u,v)\in k} (r_{uv} - p_u q_v^{\tau}) + \gamma(\|p_u\|^2 + \|q_v^{\tau}\|^2), \tag{2}$$

where $\|p_u\|$ and $\|q_v^{\tau}\|$ are the norm latent features of users and the norm of transport latent features of items, respectively.

Ensemble divide and conquer method is used to solve the rating scores' deviation that occurs when the rating scores are arranged in the memory. This method precisely learns latent features by rearranging the ratings in the rating matrix [13]. Nevertheless, the accuracy of this method is still low due to the drift in users' preferences and items' popularity decay. Thus, there is a need to study the positive effects of such temporal features for improving the prediction performance of recommendation systems [24]. Maintaining the Integrity of the Specifications.

B. Temporal Preferences

In recent years, temporal preferences and factorization factors are integrated within the collaborative-based approaches to solve sparsity problems [25]. The temporal dynamics method defines the time features by splitting the timeline into constant numbers of bins [25] while the user preferences are altered over time. This approach minimizes the overfitting of the predicted rating scores in the optimization latent space by a global weight (which is characterized by weakness in terms of personality). Generally, temporal preferences are long or short-termed [17].

- Long-term preferences

The long-term approach [17] manages the time of recorded scores to calculate the long-term preferences as expressed in Equation (3).

$$B_{uv} = \mu + B_u \frac{e - t_{uv}}{e - s} + B_u \frac{t_{uv} - s}{e - s} + B_v, \tag{3}$$

where *s* and *e* represent the first and last time preferences that are sequentially recorded in the rating matrix, t_{uv} is the current time item *v* is rated by user *u*. The long term preferences of users are defined in Equation (4) [15].

$$\vartheta_u = \exp\left[\frac{-(t_e^u - t_s^u)}{t_e^u}\right], \tag{4}$$

where ϑ_u is the temporal weight of user *u*, t_s^u is the first time and t_e^u is the last time that user *u* gave ranking scores.

Similarly, the long-term preferences of items are defined in Equation (5) [15].

$$\vartheta_v = \exp\left[\frac{-(t_e^v - t_s^v)}{t_e^v}\right], \tag{5}$$

where ϑ_v is the temporal weight of item *v*, t_s^v is the first time and t_e^v is the last time that item *v* was rated by users.

- Short-term preferences

The short-term based latent model learns the drift of users' preferences by incorporating the factorization features with the neighbors' latent feedback throughout a session, e.g., one month, one season, one year, etc. The temporal interaction model [17] combines the preferences of both short-term and long-term to address the drift in users' preferences. Nevertheless, both the short-term model and the temporal integration model [17] have limitations in terms of discovering the drift and time decay.

The short-term based factorization model [16] utilizes the *k*-means algorithm to divide the time matrix into a number of clusters based on the number of short-term periods such as the number of months. This model also deploy a bacterial foraging optimization algorithm to learn the best short-term weight to be integrated within the factorization features during the iterative learning procedure [16]. However, the sparse timestamp matrix of some active users cannot be divided into a certain number of clusters if the number of common users who rated the active user's item of interest is not sufficient (*k* problem).

In addition, longer execution time is required to learn the accurate temporal and factorization features, especially if the number of common users in the rating matrix is large. Hence, to address these issues while improving on our earlier works [1] [16], a new temporal-based approach named TWOCF is proposed. TWOCF approach assigns optimum temporal preferences and accurately learns the factorization and latent features using WOA.

C. Whale Optimization Algorithm

Whales are considered the largest animals in the world. They are always awake, quite smart, and sensitive to feelings. A truly exciting optimization idea is taken from the humpback whales, a group of whales with a unique searching technique named the bubble-net feeding method. The humpback whales only use the bubble-net for feeding. They would rather chase a bunch of krill or little fishes close to the surface. This hunting is completed by producing distinctive bubbles along a circle [26]. The feeding behaviour of humpback whales is mathematically represented by WOA to solve optimization problems.

III. Methodology

A. MovieLens Dataset

Several prediction approaches have utilized the MovieLens dataset to evaluate the performances of recommendation systems [15], [27]. All records in this dataset have been collected using the MovieLens website (movielens.umn.edu) throughout seven months from September 1997 to April 1998. There are 100,000 rating scores collected by 943 customers for 1682 movies. Each rating score that is collected by users is saved with its timestamp info. In the data set, each user has rated at least 20 movies, rating range is 1-5, each user can rate the movie as 1, 2, 3, 4, 5. The higher the user's rating of the movie, the more interested the user is in the movie. The sparseness of dataset can be calculated as: 1 – (100000/ (943 ∗ 1682)) = 0.936953. Additional features are collected by users such as age, gender, occupation, etc. Further features for items are movie id, movie title, release date, video release date, and genre. Similar to the benchmark methods, three significant features are considered in this paper. They are rating score, timestamp, and item genre.

B. Evaluation and Benchmark Methods

This work focuses on resolving the sparsity, overfitting, drift, and decay issues to improve on the prediction performance of the CF technique. The root mean square error (RMSE) function is employed for performance evaluation. RMSE has been utilized in many prediction approaches [15], [28] to evaluate the performance of the CF technique. A lower RMSE value indicates a higher prediction accuracy.

The experimental model can be benchmarked with some approaches implemented on the same dataset. These approaches are the CF, Ensemble Divide and Conquer [13], Temporal Dynamics [25], Long Temporal-based Factorization [15], Short Temporal-based Factorization [16], and Temporal-based Factorization Approach [1]. These approaches are implemented using MovieLens dataset underscoring [1-5]. Additionally, Short Temporal [16], and Temporal-based Factorization Approach [1] are implemented for three short terms which are 1 month, 2 weeks, and 1 week. The dataset contains 7 periods of 1 month, 15 periods of 2 weeks, and 30 periods of 1 week.

C. Experimental TWOCF Approach

The TWOCF method incorporates the long-term and short-term preferences with the factorization features to furnish the sparse rating matrix in order to yield accurate predictions. Besides the sparsity issue, three other challenges will be addressed by TWOCF. These are overfitting, drift, and decay as illustrated in subsequent subsections.

1) Assigning the temporal preferences: There are two kinds of temporal preferences: short-term and long-term preferences. The short temporal-based factorization model [16] analyzes the time matrix using the k-means algorithm. Similarly, the number of clusters is assigned according to the total number of sessions (e.g., MovieLens dataset contains 7 sessions and each session spans one month). However, this technique is suitable for smaller k periods (such as number of years or seasons) and not larger values (such as the number of weeks). This is the case for most users when the number of clusters formed is less than the target k after several convergences.

In addition, there are other user-centric differences. For example, some users are active only within a very short time while others are active for a longer time. These make numbering clusters in periods inaccurate. Thus, the best way to solve these challenges is by using clustering algorithms that can define the number of clusters accurately. The elbow clustering method is one of the most common methods used to determine the optimal values of clusters [29], [30]. In this work, the elbow clustering method is used to tackle the challenge of determining the number of clusters in the sparse timestamp matrix.

Fig. 1 shows a simple formation of the whale members using the elbow clustering method. Sometimes, the learning process stops because the clusters created by the k-means method are less than the required number of clusters. Equations (4) and (5) [15] are used to assign long-term preferences for users and items, respectively.

2) Integrating the temporal preferences with factorization features: This experimental work is intended to (i) predict the missing rating scores in the rating matrix following the CF technique and (ii) address other limitations such as overfitted predicted scores, drift in the users' preferences and decay in the popularity of items. The factorization and latent features that are integrated with temporal preferences are learned as follows:

$$\Re_{uv} = \mu + T_{\Omega_x^u} B_u \vartheta_u + T_{\varphi_y^v} B_v \vartheta_v + p_u q_v^\tau , \tag{6}$$

where ϑ_u and ϑ_v are the long preference of user *u* and item *v*, respectively. $T_{\Omega_x^u}$ and $T_{\varphi_y^v}$ are short-term weights assigned by cluster number *x* and cluster number *y* for user *u and* item *v*, respectively. B_u and B_v are the baseline value of user *u* and item *v* respectively. p_u is the latent feedback of user *u*, and q_v^τ is the transport latent feature of item *v*.

The WOA is used to update the short-term weights of users and items. These weights will be integrated with the factorization features and with the long-term preferences using Equation (6) to reduce the overfitted predicted rating scores. Subsequently, the WOA learns the drift in the preferences of users and the decay in the popularity of items. This helps to reduce the effects of these negative factors throughout the iterative learning process using Equations (7) and (8), respectively.

$$\Re_u = T_{\Omega_x^u} \left[B_u \vartheta_u + \sum_{u=1}^{M} (p_u)^2 \right], \tag{7}$$

$$\Re_v = T_{\varphi_y^v} \left[B_v \vartheta_v + \sum_{v=1}^{N} (q_v^\tau)^2 \right], \tag{8}$$

where *M* is the number of common users who provide rating scores for items and *N* is the number of all items rated by the active user. To obtain the best performance, this work integrates the latent feedback of Equation (6) with Equations (7) and (8) using Equation (9).

$$\widehat{\Re}_{uv} = \Re_{uv} + \Re_u + \Re_v , \tag{9}$$

where $\widehat{\Re}_{uv}$ is the predicted value for the missing rating score value by user *u* for item *v*. In Equation (9), all factors are computed by TWOCF in the first iteration and these values cannot change in other iterations except the short duration factors that can be updated throughout the optimization stages using the WOA. The predicted rating score $\widehat{\Re}_{uv}$ value can be updated throughout the iteration loops using the short-term factors $T_{\Omega_x^u}$ and $T_{\varphi_y^v}$ which provide a positive effect in improving the accuracy prediction of CF technique by reducing the error value.

3) Integrating WOA within TWOCF approach: WOA is integrated with TWOCF approach to optimize the prediction of the CF technique. This is aimed at addressing sparsity, overfitting, drift, and decay issues. TWOCF updates the weights of short temporal preferences throughout the iterative learning process managed by WOA. Three feeding behaviors of the humpback whales are briefly discussed as follows:

- Encircling prey

Humpback whales identify the locations of small fishes, then engulf them [31]. WOA algorithm assumes that the recent superlative candidate result is the objective prey or is near to the optimum. After the finest search agent is identified, the other search representatives will later attempt to revise their positions for finding the best search representative. This performance is characterized by Equations (10) and (11).

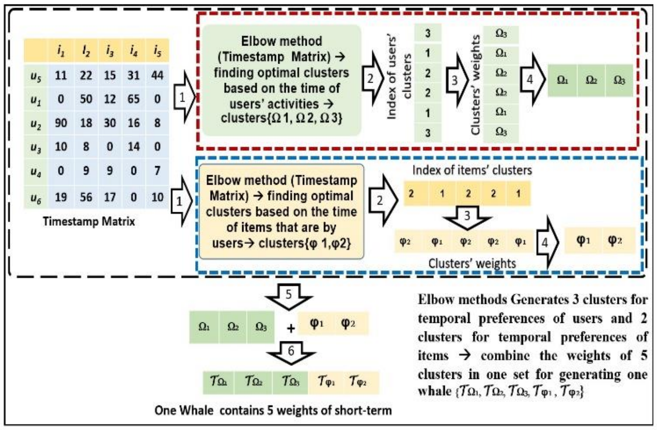
$$\vec{D} = \left| \vec{C}.\vec{T}^*(i) - \vec{T}(i) \right| , \tag{10}$$

$$\vec{T}(i+1) = \vec{T}^*(i) - \vec{A}.\vec{D} , \tag{11}$$

where *i* is the number of the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors. In this work, $\vec{T}$ represents the short duration weights which is the position vector of the prey. $\vec{T}^*$ is the position vector of the best solution (accurate short duration weights) obtained so far. The vectors $\vec{A}$ and $\vec{C}$ are computed by Equations (12) and (13) as follows:

$$\vec{A} = 2\vec{\varepsilon}.\vec{r} - \vec{\varepsilon} , \tag{12}$$

$$\vec{D} = 2.\vec{r} , \tag{13}$$



	i_1	i_2	i_3	i_4	i_5
u_5	11	22	15	31	44
u_1	0	50	12	65	0
u_2	90	18	30	16	8
u_3	10	8	0	14	0
u_4	0	9	9	0	7
u_6	19	56	17	0	10

Timestamp Matrix
1
Elbow method (Timestamp Matrix) → finding optimal clusters based on the time of users' activities → clusters{Ω 1, Ω 2, Ω 3}
2
Index of users' clusters
3 1 2 2 1 3
3
Clusters' weights
Ω3 Ω1 Ω2 Ω2 Ω1 Ω3
4
Ω1 Ω2 Ω3
1
Elbow method (Timestamp Matrix) → finding optimal clusters based on the time of items that are by users→ clusters{φ 1,φ2}
2
Index of items' clusters
2 1 2 2 1
3
φ2 φ1 φ2 φ2 φ1
Clusters' weights
4
φ1 φ2
5
Ω1 Ω2 Ω3 + φ1 φ2
6
TΩ1 TΩ2 TΩ3 Tφ1 Tφ2
One Whale contains 5 weights of short-term
Elbow methods Generates 3 clusters for temporal preferences of users and 2 clusters for temporal preferences of items → combine the weights of 5 clusters in one set for generating one whale {TΩ1, TΩ2, TΩ3, Tφ1 , Tφ2}

Fig. 1. A Simple Example of Forming a Whale's Members.

where $\vec{\varepsilon}$ is linearly reduced from 2 to 0 throughout the iterative process and $\vec{r}$ is a random vector from 0 to 1. Equation (11) allows each search agent to update the temporal weight that is close to the current best position and replicate circling the prey.

- Bubble-net attacking method

WOA mathematical modeling involves two phases modeled as follows:

1) Shrinking encircling mechanism: In Equation (12), the value of $\vec{\varepsilon}$ is reduced, then $\vec{A}$ is also reduced by $\vec{\varepsilon}$. Therefore, the new position of a search agent can be defined between the current temporal positions and the best temporal positions by setting random values for $\vec{A}$ from 0 to 1.

2) Spiral updating position: The distance between the whale and its prey can be calculated using the weights of temporal features and the best weights of temporal features. A spiral equation is then formed between the position of whale and prey as shown in Equation (14).

$$\vec{T}(i+1) = \vec{D}'.\exp(b\omega).\cos(2\pi\omega) + \vec{T}^{*}(i), \tag{14}$$

where $\vec{D}' = \left|\vec{T}^{*}(i) - \vec{T}(i)\right|$ denotes the distance between the whale and its prey, *b* is a constant number to define the form of logarithmic rise, and ω is a random number in [-1, 1]. The humpback whales swim across the prey contained by a disappearing circle along a spiral-shaped path concurrently. This simultaneous behavior takes a 50% probability to choose either the shrinking surrounding structure (Equation (6)) or the spiral model for updating the temporal weights (Equation (9)) during the optimization as shown below.

$$\begin{aligned}
&if\,(p < 0.5)\\
&\vec{T}(i+1) = \vec{D}'.\exp(b\omega).\cos(2\pi\omega) + \vec{T}^{*}(i)\\
&elseif\,(p >= 0.5)\\
&\vec{T}(i+1) = \vec{T}^{*}(i) - \vec{A}.\vec{D}
\end{aligned}$$

where *p* is a random number in [0,1].

- Search for prey (Exploration phase)

In this situation, $\vec{A}$ is utilized by the random values greater than 1 or lesser than −1 for forcing the search agent to change the current whale's location [31]. This allows the WOA algorithm to perform a global search that is modeled using Equations (15) and (16).

$$\vec{D} = \left|\vec{C}.\vec{T}_{rand} - \vec{T}\right|, \tag{15}$$

$$\vec{T}(i+1) = \vec{T}_{rand} - \vec{A}.\vec{D}, \tag{16}$$

where $\vec{T}_{rand}$ is a random temporal vector. The phases of TWOCF approach are further detailed in the TWOCF algorithm.

TWOCF algorithm

```
  Procedure of Factorization:
    Input: Rating Matrix
    Output: $\mu, B_u, B_v, p_u, q_v, q_v^{\tau}$
      $B_u = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(r_{uv}-\mu)^2}$
      $B_v = \sqrt{\frac{1}{m}\sum_{v=1}^{m}(r_{uv}-\mu)^2}$
      $[p_u, d, q_v] = SVD(RatingMatrix)$
      $q_v^{\tau}$ =transpose ( $q_v$ )
Procedure of Long-term:
  Input: Timestamp Matrix
  Output: $\vartheta_u, \vartheta_v$
      $\vartheta_u = \exp\left[-(t_e^u - t_s^u)/t_e^u\right]$
      $\vartheta_v = \exp\left[-(t_e^v - t_s^v)/t_e^v\right]$
Procedure of Short-Term:
  Input: Timestamp Matrix
  Output: $[T_{\Omega}, T_{\Omega_2}, T_{\Omega_3}, \ldots\ldots, T_{\Omega_x}, T_{\varphi_1}, T_{\varphi_1}, T_{\varphi_1}, \ldots\ldots\ldots, T_{\varphi_y}]$
      Clustering users
        [Index^u, x] = Elbow clustering Method
        (Timestamp Matrix)
        $T_{\Omega_1}, T_{\Omega_2}, T_{\Omega_3}, \ldots\ldots, T_{\Omega_x}$ = [Index^u, x]
      Clustering items
        [Index^v, y] = Elbow clustering Method
        (Timestamp Matrix)
        $T_{\varphi_1}, T_{\varphi_1}, T_{\varphi_1}, \ldots\ldots\ldots, T_{\varphi_y}$ = [Index^v, y]
      Managed weights of users and items in one
      whale members
        weights=
        $[T_{\Omega}, T_{\Omega_2}, T_{\Omega_3}, \ldots\ldots, T_{\Omega_x}, T_{\varphi_1}, T_{\varphi_1}, T_{\varphi_1}, \ldots\ldots\ldots, T_{\varphi_y}]$
Procedure of Collaborative Filtering (Fitness):
  Input: Rating Matrix, Timestamp Matrix
  Output: RMSE value
      Call Procedure of Factorization
      $\mu, B_u, B_v, p_u, q_v, q_v^{\tau}$
      Call Procedure of Long-term to obtain $\vartheta_u, \vartheta_v$
      Call Procedure of WOA to update
      $[T_{\Omega}, T_{\Omega_2}, T_{\Omega_3}, \ldots\ldots, T_{\Omega_x}, T_{\varphi_1}, T_{\varphi_1}, T_{\varphi_1}, \ldots\ldots\ldots, T_{\varphi_y}]$
      Predicting the missing values in Rating Matrix
        $\mathfrak{R}_{uv} = \mu + T_{\Omega_x^u} B_u \vartheta_u + T_{\varphi_y^v} B_v \vartheta_v + p_u q_v^{\tau}$
        $\mathfrak{R}_u = T_{\Omega_x^u}\left[B_u \vartheta_u + \sum_{u=1}^{M}(p_u)^2\right]$
        $\mathfrak{R}_v = T_{\varphi_y^v}\left[B_v \vartheta_v + \sum_{v=1}^{N}(q_v^{\tau})^2\right]$
        $\widehat{\mathfrak{R}}_{uv} = \mathfrak{R}_{uv} + \mathfrak{R}_u + \mathfrak{R}_v$
      Getting similarity Values using Cosine
      Function
        $sim(u_a, u_c) = \sum_{v=1}^{M}(r_{u_a,item_v}.r_{u_c,item_v}) \Big/ \left[\sqrt{\sum_{v=1}^{M} r_{u_a,item_v}^2}.\sqrt{\sum_{v=1}^{M} r_{u_c,item_v}^2}\right]$
      Getting predicted Values for the items rated by
      active user
```

```
        $p_v = \mu_{u_a} + \left[ \sum_{c=1}^{M} sim(u_a,u_c)(r_{u_c,item_v} - \mu_{u_c}) \right] \Big/ \sqrt{\sum_{c=1}^{M} |sim(u_a,u_c)|}$
        ($p_v$ is real/infinite value in [0-5])
      Getting fitness Value by RMSE function
        $RMSE_{u_a} = \sqrt{\frac{1}{N}\sum_{v=1}^{N}(r_v - p_v)^2}$

Procedure of WOA:
    Input: $[T_{\Omega},T_{\Omega_2},T_{\Omega_3},......,T_{\Omega_x},T_{\varphi_1},T_{\varphi_1},T_{\varphi_1},.......,T_{\varphi_y}]$
    Output: updated $[T_{\Omega},T_{\Omega_2},T_{\Omega_3},.....,T_{\Omega_x},T_{\varphi_1},T_{\varphi_1},T_{\varphi_1},.......,T_{\varphi_y}]$
      Assign the parameters of WOA
      $\vec{\varepsilon} = random(between 2 \,\&\&0)$
      $\vec{r} = random(between 0 \,\&\&1)$
      $\vec{A} = 2\vec{\varepsilon}.\vec{r} - \vec{\varepsilon}$
      $\vec{D} = 2.\vec{r}$
      Search agent (whale)=10
      T*=the best temporal weights (the best search agent)
      While iteration <MaxIteration
        For each search agent
        $[T_{\Omega},T_{\Omega_2},T_{\Omega_3},......,T_{\Omega_x},T_{\varphi_1},T_{\varphi_1},T_{\varphi_1},.......,T_{\varphi_y}]$
        $\vec{D} = |\vec{C}.\vec{T}_{rand} - \vec{T}|$
        Create a random number in [0-1] → p
        $if\,(p < 0.5)$
        $\vec{T}(i+1) = \vec{D}'.\exp(b\omega).\cos(2\pi\omega) + \vec{T}^*(i)$
        $elseif\,(p >= 0.5)$
        $\vec{T}(i+1) = \vec{T}^*(i) - \vec{A}.\vec{D}$
        Call Procedure of Collaborative Filtering (Fitness) → $RMSE_{u_a}$
        if RMSE<fitness
         fitness=RMSE
         update $[T_{\Omega},T_{\Omega_2},T_{\Omega_3},...,T_{\Omega_x},T_{\varphi_1},T_{\varphi_1},T_{\varphi_1},....,T_{\varphi_y}]$
        end if
      End While
Main TWOCF
  Data preparation based on personalized
      Assign the active user $u_a$
      Arrange scores value in Rating Matrix
      Arrange timestamp value in Timestamp Matrix
        Active user is arranged in the first row and the items rated by active user are arranged as columns
        Add any user provides scores to any item rated by active user to the Matrix
         → Rating Matrix, Timestamp Matrix
  Procedure of Short-Term
   $[T_{\Omega},T_{\Omega_2},T_{\Omega_3},......,T_{\Omega_x},T_{\varphi_1},T_{\varphi_1},T_{\varphi_1},.......,T_{\varphi_y}]$
      Assign WOA vectors
        Search Agents= 10
        dim=x+y → (number of short-term weights)
        The upper bound of each weight in dim = 1
        The lower bound of each weight in dim = -1
        MaxIteration=300
        Initialize random values for whale members →
         Matrix (Search Agents, dim) =rand
        Repeat
          Call Procedure of WOA → RMSE
            If fitness<RMSE
              Fitness=RMSE
              Update $[T_{\Omega},T_{\Omega_2},T_{\Omega_3},......,T_{\Omega_x},T_{\varphi_1},T_{\varphi_1},T_{\varphi_1},.......,T_{\varphi_y}]$
        Until complete Max iteration
    Output:
    The accurate Rating Matrix with predicted missing scores
    The accurate list of recommendations for the active user
```

IV. Experimental Results

A. The Effect of TWOCF in Solving the Overfitting, Drift, and Decay

The TWOCF approach is proposed to address the weaknesses of factorization and temporal-based factorization approaches. Experimental results show that the TWOCF approach exhibits significantly superior performance with respect to learning the accurate temporal and factorization features by reducing overfitted predicted scores, tracing decay in the popularity of items, and tracing drifting in the users' preferences. Table I presents the experimental results obtained after implementing the TWOCF approach on MovieLens dataset under-scoring [1-5].

The first column in Table I contains 31 active users. The second and third columns indicate the dimensions of the rating matrix for assigning the learning search space. Columns 4 and 5 show the cluster's number based on users' and items' dimensions, respectively. The different numbers of clusters in each matrix refer to personality behaviors of users. It is worthy of note that this cannot be learned accurately using a specific number of clusters. The sixth column shows several numbers of whale members by which the TWOCF approach accurately learns the features of users and items.

In column 7, the execution time of learning procedures varies according to the dimension space of each matrix. The shortest execution time is 11 seconds while the longest execution time is 487 seconds. The last column indicates the prediction accuracy of the CF technique according to RMSE values. Here, a lower value indicates a higher prediction accuracy. Using the TWOCF approach, results ranging from 0.523 to 0.997 with an average of 0.764 are obtained.

The learning processes by the TWOCF approach are visualized in Fig. 2 to show its ability to reduce the RMSE values throughout the iteration loops. Fig. 2 shows the effectiveness of the TWOCF approach in accurately learning the behaviours of users and items throughout the learning iteration. This improves the CF technique's speed and learning accuracy.

B. Comparison of the Performances of CF, Factorization, and Factorization-based Temporal Approaches

Here, the TWOCF approach is evaluated by comparing its effectiveness in reducing the RMSE value with other benchmark approaches described in Section 2. The TWOCF and benchmark approaches compared are implemented using one Test-Set (contains 31 rating matrices) to predict the missing scores in each rating matrix. In addition, the contributions of the tested approaches to solve the issues that are reviewed in this article are summarized in Table II.

TABLE I. THE EXPERIMENTAL RESULTS OF THE TWOCF APPROACH

Active User	Matrix Dimensions		Clusters numbers		Whale members	Execution Time (Seconds)	RMSE value
	Common Users	Items	Common Users	Items			
u1	937	43	24	7	31	21	0.994
u2	928	207	21	14	35	124	0.723
u3	943	299	21	15	36	205	0.784
u4	920	26	24	6	30	13	0.979
u5	916	32	25	6	31	15	0.541
u6	883	63	21	7	28	26	0.693
u7	939	132	22	11	33	64	0.740
u8	871	24	24	5	29	11	0.670
u9	943	138	23	11	34	70	0.802
u10	876	20	26	5	31	10	0.885
u11	914	147	22	12	34	84	0.872
u12	942	102	23	11	34	50	0.665
u13	931	31	24	6	30	17	0.628
u14	930	36	24	6	30	18	0.555
u15	943	539	23	20	43	487	0.678
u16	938	60	22	8	30	27	0.829
u17	884	30	24	6	30	14	0.680
u18	943	63	24	8	32	29	0.738
u19	875	22	23	5	28	11	0.845
u20	840	89	18	10	28	37	0.826
u21	940	108	21	9	30	52	0.890
u22	941	224	22	11	33	148	0.944
u23	885	115	24	11	35	52	0.849
u24	845	30	26	6	32	13	0.882
u25	895	33	26	6	32	16	0.731
u26	938	55	23	8	31	26	0.523
u27	890	26	25	6	31	13	0.638
u28	933	196	23	13	36	124	0.722
u29	943	269	23	16	39	184	0.679
u30	911	45	23	7	30	20	0.713
u31	941	63	23	8	31	29	0.997
Average	915	105	23	9	32	65	0.764

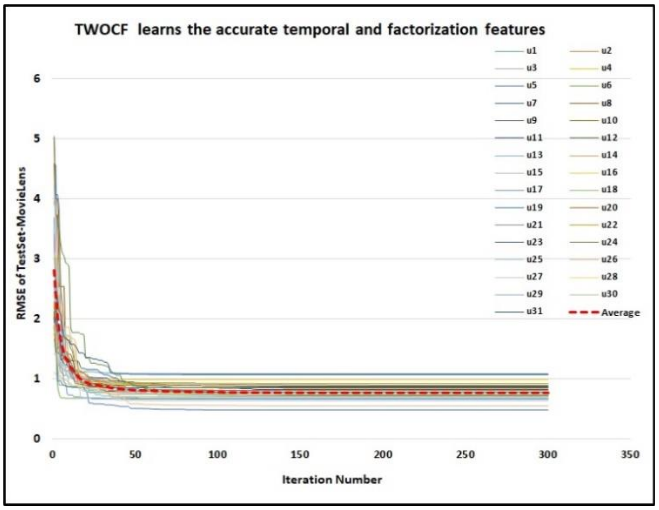


Fig. 2. TWOCF Improves the Prediction Accuracy of the CF Technique.

The CF technique has provided the lowest accuracy prediction because of the negative effects of sparsity, drift, and decay issues. The highest RMSE value represents the lowest accuracy prediction. Ensemble Divide and Conquer [13] is used to solve the sparsity and missing the accurate location of data when arrange this data into memory. The output results show better performance compared to the CF. However, this method has a weakness in terms of drift and decay. Long Temporal based Factorization [15] is used to solve the sparsity, overfitting and decay issues by learning the long-term features through the convergence of genres features of the items.

TABLE II. THE CF PERFORMANCE USING THE PREDICTION APPROACHES

Approach	Objectives				Long duration	Short duration assigned by	Execution time (Seconds)	Average RMSE
	Sparsity	Overfitting	Drift	Decay				
CF								0.957
Ensemble Divide and Conquer [13]	√							0.954
Temporal Dynamics [25]	√	√	√	√	√	Time slices		0.951
Long Temporal based Factorization [15]	√	√		√	√			0.947
Short Temporal based Factorization [16]	√	√	√			1 week	4256	0.851
						2 weeks	4049	0.863
						1 month	3921	0.870
Temporal based Factorization [1]	√	√	√	√	√	1 week	1316	**0.818**
						2 weeks	4128	0.819
						1 month	4197	0.843
TWOCF	√	√	√	√	√	Auto clustering	**65**	**0.764**

The short temporal features (column 7) are defined by different duration factors to achieve accurate solutions. For example, Temporal Dynamics [25] defines the short-term by time slices. However, it has weaknesses in terms of personality.

Short Temporal-based Factorization [16] and Temporal-based Factorization [1] approaches defined the short-term periods using the k-means algorithm where the number of the clusters is assigned based on the number of certain times (e.g., in MovieLens the whole time of users' activities can be divided into 7, 15, or 30 clusters when assigning one month, 2 weeks, or one week period, respectively).

The Short Temporal-based Factorization [16] is used to solve drift of users and ignored the issue of decay during long duration which reduce the accuracy prediction performance of the CF. Temporal-based Factorization [1] approach is used to solve all issues. Its result is the best comparing to the benchmark approaches. However, the benchmark approaches are ignored the results of running time due to the iteration of optimization procedure is very slow as shown in Table II (e.g. minimum average of running time is 1316 second). Bacterial foraging optimization algorithm is used with the last three benchmark approaches of temporal and its experimental running times are slow as shown in Table II which represent as a significant weakness. The recommendation systems need high accuracy as well as faster running time. As observed in Table II, the studied approaches have different executing time and accuracy. It is obvious that approaches with high accuracy have long run time, e.g., Temporal-based Factorization [1] provides lower RMSE (high accuracy) but with a long executing time.

Distinctively, TWOCF approach learns the accurate features of each user within the smallest execution time. Additionally, the TWOCF approach provides the highest accuracy prediction compared to the other benchmark approaches. This means that the TWOCF approach has the best performance and can deal with all kinds of matrices as the number of clusters is assigned automatically. Moreover, TWOCF approach performs best in reducing the overfitted predicted scores and accurately learning the temporal features throughout the learning iteration, which reducing the negative effect of drift and decay in the prediction performance of the CF technique.

V. Conclusion

Recommendation systems are becoming popular because they can efficiently recommend products to customers based on their interests. CF-based recommendation systems perform well since they consider the rating matrix in their execution. Nevertheless, CF suffers from the sparsity issue which is usually tackled using factorization approaches. Similarly, overfitting is another challenge mainly addressed using optimization approaches. Additionally, the drift in users' preferences and items' popularity decay addressed by Temporal-based factorization approaches are also major setbacks. Although the current solutions achieve some level of accuracy, there is still room for improvement. For example, dividing the temporal activities throughout the duration search space, reducing the runtime of the execution process, and lowering the error values of the predicted rating scores.

The TWOCF approach is proposed to render timely and accurate predictions within the rating matrix by accurately learning users' preferences and items' popularity pattern. TWOCF adopts the elbow clustering method to obtain the optimal number of temporal clusters. Also, the short-term weights of generated clusters are integrated with the factorization features for predicting the missing scores in the rating matrix. Results show that the TWOCF approach outperforms the benchmark schemes, improves the accuracy of the CF technique, and reduces its execution time.

References

[1] A. A.-Q. Al-Hadi, N. Mohd Sharef, S. M. Nasir, and M. Norwati, "Temporal based factorization approach for solving drift and decay in sparse scoring matrix," Int. Conf. Soft Comput. Data Mining, Springer, vol. 700, pp. 340–350, 2018.

[2] M. Ranjbar, P. Moradi, M. Azami, and M. Jalili, "An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems," Eng. Appl. Artif. Intell., vol. 46, pp. 58–66, 2015.

[3] M. H. Mohamed, M. H. Khafagy, and M. H. Ibrahim, "Recommender Systems Challenges and Solutions Survey," Int. Conf. Innov. Trends Comput. Eng., pp. 149–155, 2019.

[4] D. Chae, J. Kim, and S. Kim, "AR-CF : Augmenting Virtual Users and Items in Collaborative Filtering for Addressing Cold-Start Problems," Sigir 2020, no. July, pp. 1251–1260, 2020.

[5] Z. Zhang, Y. Kudo, T. Murai, and Y. Ren, "Improved covering-based collaborative filtering for new users' personalized recommendations," Knowl. Inf. Syst., vol. 62, no. 8, pp. 3133–3154, 2020.

[6] B. Alhijawi and Y. Kilani, "The recommender system: A survey," Int. J. Adv. Intell. Paradig., vol. 15, no. 3, pp. 229–251, 2020.

[7] A. Pujahari and D. S. Sisodia, "Pair-wise Preference Relation based Probabilistic Matrix Factorization for Collaborative Filtering in Recommender System," Knowledge-Based Syst., vol. 196, no. xxxx, p. 105798, 2020.

[8] M. Nilashi et al., "Preference learning for eco-friendly hotels recommendation: A multi-criteria collaborative filtering approach," J. Clean. Prod., vol. 215, pp. 767–783, 2019.

[9] I. A. A. Q. Al-Hadi, N. M. Sharef, M. N. Sulaiman, and N. Mustapha, "Review of the temporal recommendation system with matrix factorization," Int. J. Innov. Comput. Inf. Control, vol. 13, no. 5, pp. 1579–1594, 2017.

[10] N. Zainal, I. A. A. Q. Al-Hadi, S. M. Ghaleb, H. Hussain, W. Ismail, and A. Y. Aldailamy, "Predicting MIRA Patients' Performance Using Virtual Rehabilitation Programme by Decision Tree Modelling," Recent Adv. Intell. Syst. smart Appl. Cham Springer, pp. 451–462, 2020, doi: 10.1007/978-3-030-47411-9_24.

[11] A. Al-badarneh, M. Ali, and S. M. Ghaleb, "An Improved Classifier for Arabic," J. Converg. Inf. Technol., vol. 11, no. 3, pp. 69–84, 2016.

[12] I. A. A. Al-hadi, N. M. Sharef, N. Mustapha, and M. Nilashi, "Latent based temporal optimization approach for improving the performance of collaborative filtering," PeerJ Comput. Sci., pp. 1–25, 2020, doi: 10.7717/peerj-cs.331.

[13] I. A. A. Q. Al-Hadi, N. M. Sharef, M. N. Sulaiman, and N. Mustapha, "Ensemble divide and conquer approach to solve the rating scores' deviation in recommendation system," J. Comput. Sci., vol. 12, no. 6, pp. 265–275, 2016.

[14] M. Guo, J. Sun, and X. Meng, "A Neighborhood-based Matrix Factorization Technique for Recommendation," Ann. Data Sci., vol. 2, no. 3, pp. 301–316, 2015.

[15] I. A. A.-Q. Al-Hadi, N. M. Sharef, M. N. Sulaiman, and N. Mustapha, "Temporal-Based Approach to Solve Item Decay Problem in Recommendation System," Adv. Sci. Lett., vol. 24, no. 2, pp. 1421–1426, 2018.

[16] I. A. A.-Q. Al-Hadi, N. M. Sharef, M. N. Sulaiman, and N. Mustapha, “Bacterial foraging optimization algorithm with temporal features to solve data sparsity in recommendation system,” ICC ’17 Proc. Second Int. Conf. Internet things, Data Cloud Comput. ACM., pp. 148:1--148:6, 2017.

[17] F. Ye and J. Eskenazi, “Feature-Based Matrix Factorization via Long- and Short-Term Interaction,” Knowl. Eng. Manag., pp. 473–484, 2014.

[18] Y. Koren, “Factorization meets the neighborhood: A multifaceted collaborative filtering model,” Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 426–434, 2008.

[19] D. Yang, T. Chen, W. Zhang, and Y. Yu, “Collaborative filtering with short term preferences mining,” Proc. 35th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR ’12, no. 2, p. 1043, 2012.

[20] “Sebastien De Landtsheer (2020). kmeans_opt (https://www.mathworks.com/matlabcentral/fileexchange/65823-kmeans_opt),” MATLAB Cent. File Exch. Retrieved Sept. 3, 2020.

[21] R. Lara-Cabrera, Á. González-Prieto, angel gonzalez prieto@upm Es, F. Ortega, and jesus bobadilla@upm es Jesús Bobadilla, “Evolving matrix-factorization-based collaborative filtering using genetic programming,” Appl. Sci., vol. 10, no. 2, 2020.

[22] M. Nilashi, O. Ibrahim, and K. Bagherifard, “A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques,” Expert Syst. Appl., vol. 92, pp. 507–520, 2018.

[23] X. Liu, J. Zhang, and C. Yan, “Towards context-aware collaborative filtering by learning context-aware latent representations,” Knowledge-Based Syst., vol. 199, pp. 1–13, 2020.

[24] J. Vinagre, A. M. Jorge, and J. Gama, “An overview on the exploitation of time in collaborative filtering,” Wiley Interdiscip. Rev. Data Min. Knowl. Discov., vol. 5, no. 5, pp. 195–215, 2015.

[25] Y. Koren, “Collaborative filtering with temporal dynamics,” Commun. ACM, vol. 53, no. 4, pp. 89–97, 2010.

[26] S. Adhirai, R. P. Mahapatra, and P. Singh, “The Whale Optimization Algorithm and Its Implementation in MATLAB,” Int. J. Comput. Inf. Eng., vol. 12, no. 10, pp. 815–822, 2018.

[27] H. Koohi and K. Kiani, “A new method to find neighbor users that improves the performance of Collaborative Filtering,” Expert Syst. Appl., vol. 83, pp. 30–39, 2017.

[28] F. Pajuelo-Holguera, J. A. Gómez-Pulido, F. Ortega, and J. M. Granado-Criado, “Recommender system implementations for embedded collaborative filtering applications,” Microprocess. Microsyst., vol. 73, 2020.

[29] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, “Integration K-Means Clustering Method and Elbow Method for Identification of the Best Customer Profile Cluster,” IOP Conf. Ser. Mater. Sci. Eng., vol. 336, no. 1, 2018.

[30] P. Bholowalia and A. Kumar, “EBK-means: a clustering technique based on elbow method and K-Means in WSN,” Int. J. Comput. Appl., vol. 105, no. 9, pp. 17–24, 2014.

[31] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” Adv. Eng. Softw., vol. 95, pp. 51–67, 2016.