Machine Learning with Applications 14 (2023) 100495

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

A collaborative filtering recommendation framework utilizing social networks

Aamir Fareed [a], Saima Hassan [a], Samir Brahim Belhaouari [b,*], Zahid Halim [c]

[a] Institute of Information Technology, Kohat University of Science and Technology, Kohat, 26000, Pakistan
[b] Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha, P.O. Box 3411, Qatar
[c] Faculty of Computer Science and Engineering, GIK Institute of Science and Technology, Topi, 23460, Pakistan

ARTICLE INFO

Keywords:
Recommendation systems
Collaborative filtering
Social networks
Data sparsity

ABSTRACT

Collaborative filtering is a widely used technique for providing personalized recommendations to users. However, traditional collaborative filtering methods fail to consider the social connections between users. The current study proposes a collaborative filtering recommendation framework that employs social networks to generate more precise and pertinent recommendations. The framework is based on a modified version of the user-based collaborative filtering algorithm, which computes user similarity based on their ratings and social connections. The similarity measure is determined by a weighted combination of these two factors, with the weights learned through an optimization process. The framework is evaluated using a dataset of movie ratings and social connections between users. The findings reveal that the proposed approach outperforms traditional collaborative filtering methods regarding recommendation accuracy and relevance. Moreover, the framework can offer more diverse recommendations compared to traditional methods. In summary, the proposed framework integrates social networks to enhance the accuracy and relevance of collaborative filtering recommendations. The approach has various applications, including e-commerce, music, and movie recommendation, and can potentially address the issues of cold-start and sparsity in collaborative filtering.

1. Introduction

In recent years, personalized recommendation systems have gained significant attention due to their ability to provide users with tailored recommendations based on their preferences and behavior (Rajesh et al., 2019). Collaborative filtering is a popular method recommendation systems use to provide personalized recommendations to users (Shahbazi et al., 2020). Traditional collaborative filtering (CF) methods rely solely on the similarity between users' preferences, as inferred from their past behaviors or ratings, to generate recommendations (Paradarami et al., 2017). However, these methods ignore the valuable information contained in social networks that connect users. Social networks have become an essential part of our daily lives, and the connections we form with others can reveal valuable information about our preferences, interests, and behaviors (Priambodo et al., 2019). Incorporating social network information into CF methods can lead to more accurate and relevant recommendations. By utilizing social connections, the recommendation system can leverage the implicit feedback a user's network of friends provides, such as their likes, shares, and comments (Tahmasebi et al., 2021).

The proposed framework of this research is based on a modified version of the user-based CF algorithm (Shen et al., 2020). The similarity between the users is calculated based on both ratings and social connections of the user, using a weighted combination of these two factors to determine the similarity (Yao et al., 2018). The proposed framework is evaluated using a dataset of movie ratings and social connections between users and compares the performance with the traditional CF methods (Li et al., 2019). The proposed framework has several potential applications in various domains such as e-commerce, music, and movie recommendation. One of the significant advantages of the framework is its ability to address the cold-start problem and sparsity issues, which are common challenges faced by recommendation systems. A recommendation system is of three types: content-based, CF, and a hybrid recommendation system which can be depicted in Fig. 1. Since the focus of this research is a CF recommendation system (Chang et al., 2016); therefore, it will be discussed in detail.

In addition to the specific literature review done on the CF technique, this research also contributes by presenting a recommendation framework that utilizes social network information for the improvement of the accuracy and diversity of recommendations made by CF

* Corresponding author.
E-mail addresses: cs001af@kust.edu.pk (A. Fareed), saimahassan@kust.edu.pk (S. Hassan), sbelhaouari@hbku.edu.qa (S.B. Belhaouari), Zahid.halim@giki.edu.pk (Z. Halim).

https://doi.org/10.1016/j.mlwa.2023.100495
Received 11 January 2023; Received in revised form 24 August 2023; Accepted 25 August 2023
Available online 4 September 2023
2666-8270/

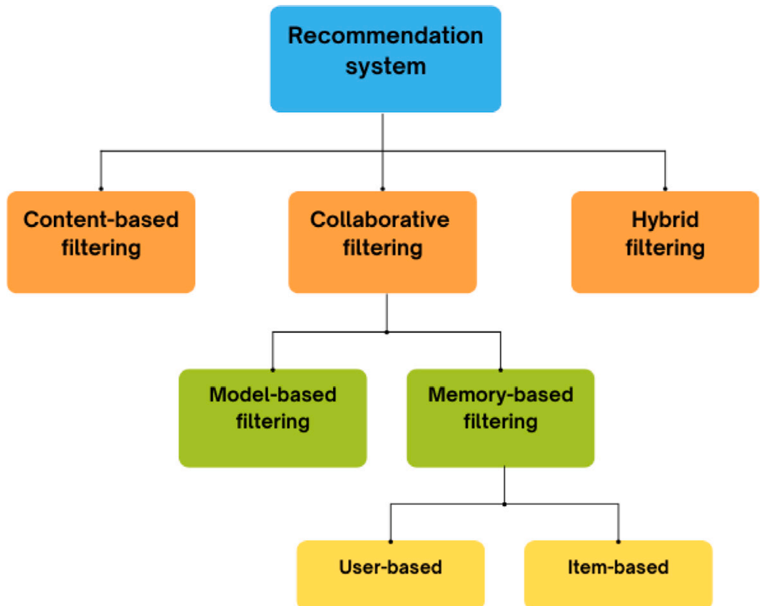

Fig. 1. Overview of recommendation system.

algorithms. The objective is to overcome the sparsity issue that traditional CF methods have often faced by incorporating the preferences of a user's friends into the recommendation process. This research aims to demonstrate that the proposed approach can lead to more accurate and diverse recommendations for users and evaluate its effectiveness through experiments and user studies.

The rest of the paper is organized as follows: Section 2 provides a background study of the CF recommendation system. Section 3 presents the proposed methodology for addressing various challenges in the CF recommendation system. In Section 4, the results obtained from the proposed model are discussed. Finally, Section 5, Concludes the significance as well as the development of the CF recommendation system for further research.

2. Collaborative filtering recommendation system

Collaborative filtering (CF) is a widely used technique in recommendation systems that aims to provide users with personalized recommendations based on their past behavior and preferences (Chang et al., 2016). However, traditional CF methods have limitations, particularly when it comes to dealing with the cold-start problem and sparsity in the dataset (Shrestha & Yang, 2019). One approach to address these issues is to incorporate social connections between users into the CF algorithm. Social connections can provide additional information about users, such as their interests, preferences, and behavior, that can be used to enhance the recommendation accuracy and relevance (Zhang et al., 2020). A study by Madani et al. (2019) proposed a social CF algorithm that integrates the social network information into the CF model. The algorithm utilized the social network structure to calculate the similarity between users and provided more accurate recommendations compared to traditional CF algorithms. Similarly, Fan et al. (2019) proposed a social recommendation framework that utilizes both the user–item interaction and social network information to make recommendations. The framework was evaluated on a dataset of movie ratings and social network information and showed significant improvements in recommendation accuracy and diversity. Another approach to incorporating social information into recommendation systems is through social tagging. Social tagging involves allowing users to tag items with descriptive keywords or phrases, which can be used to capture the user's interests and preferences (Naseri et al., 2015). A study in Huang (2019) proposed a social tagging-based recommendation method that integrates social tagging data into the CF algorithm. The method showed promising results in improving recommendation accuracy and alleviating the cold-start problem.

An in-depth explanation of the latent factor (LFM) model and its current parameters, including matrix factorization, nonlinear matrix factorization, and singular value decomposition (SVD), are the most widely used recommendation approaches for the use of CF recommendation systems (Zeng et al., 2017). Memory-based and model-based techniques are both employed in classical cognitive facilitation, which is a kind of CF technique (Zhang et al., 2021). The CF recommendation system's framework is depicted in Fig. 2.

2.1. Memory-based collaborative filtering

The memory-based CF method, which itself focused on the active user's object rating matrix, can be used to link people and items as shown in Fig. 3. The method suggests highly-rated items for people who are similar to the active user (Fessahaye et al., 2019). User ratings are used to predict new items in a CF recommendation system with no requirement for further data or data collection. User-based CF and Object-based CF are the two categories of memory-based recommendation techniques that were utilized in Raza and Ding (2019). Fig. 3 depicts an illustration of the relationship between user-based and object-based cognitive flexibility.

2.1.1. User-based collaborative filtering

User-based CF, which is based on the premise that users who have similar prior ratings should likewise have similar interests, is gaining popularity and allows us to anticipate active user ratings that are not tied to specific commodities by analyzing prior ratings (Koohi & Kiani, 2021). The active user's neighbors are identified based on the comparison of the active user and other users, which is done before identifying who the active user's neighbors are. Until the neighbors of the current user are decided, this process is repeated. The ratings that an active user receives, in turn, are based on popular past information from the same adjacent users, and the suggestion results are generated based on the popular past information (Li et al., 2017).

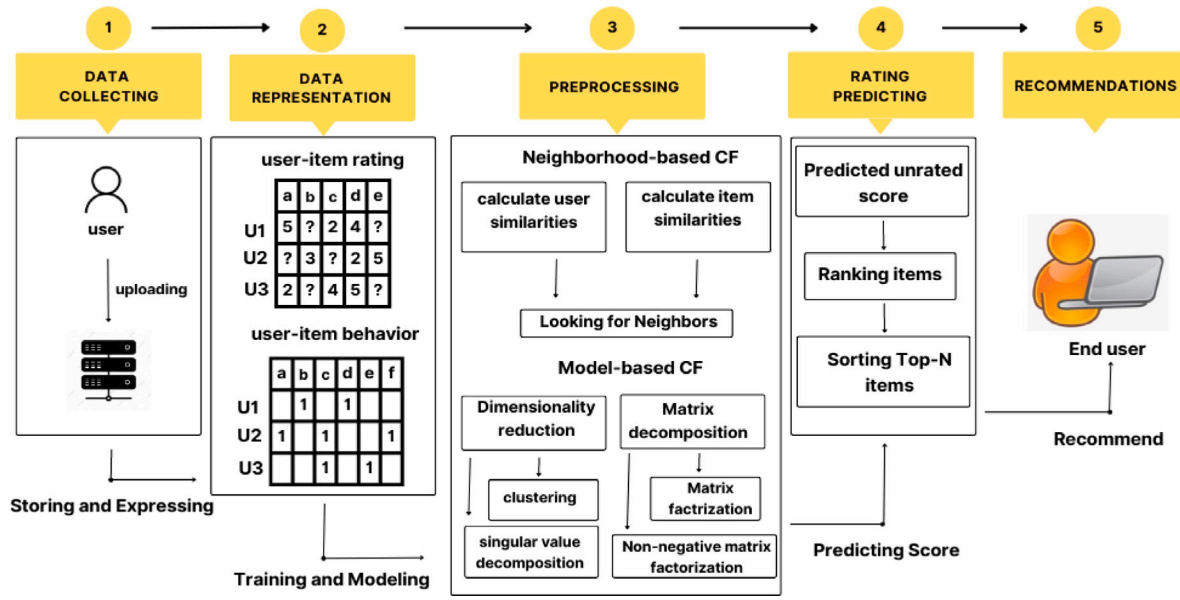

Fig. 2. Data flow in collaborative filtering recommendation system.

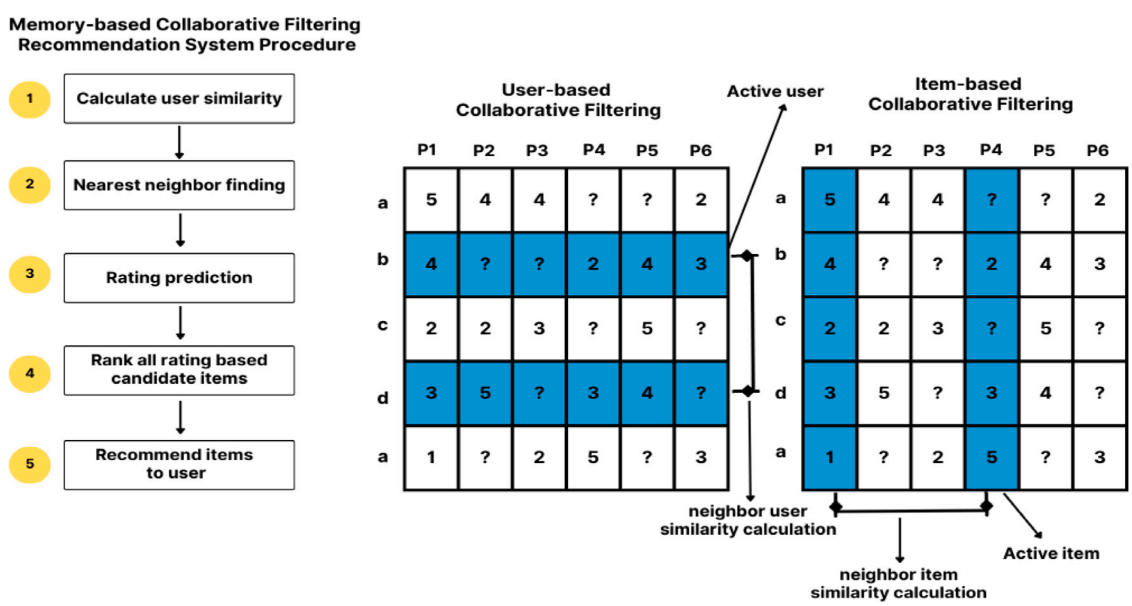

Fig. 3. Rationale of user and item-based collaborative filtering.

1. **Calculating users similarity**
The rating vector $r_u = \{r_{u1}, r_{u2}, \ldots, r_{un}\}$, which reflects the user u ratings in three dimensions, is extensively used in computer science and is seen in the ratings of the user u. The degree to which two users are similar can be determined by comparing the ratings of their respective rating vectors. It has traditionally been used to determine how closely users are related to one another using the cosine similarities metric and the Pearson correlation coefficient (PCC) (Koohi & Kiani, 2021). N-dimensional vectors can be constructed from user ratings to represent their similarity, and this can be used to determine how closely two users' ratings resemble each other. According to a common rule of thumb, the more similar two items are, the smaller the angle between them is calculated as in Koohi and Kiani (2021). Eq. (1) is used to determine the cosine vector similarity between two vectors.

$$sim_{uv} = cos(\vec{r}_u, \vec{r}_v) = \frac{\vec{r}_u.\vec{r}_v}{\|\vec{r}_u\|_2 \times \|\vec{r}_v\|_2} = \frac{\sum_{i \in I_{uv}} \vec{r}_{ui}, .\vec{r}_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2}\sqrt{\sum_{i \in I_v} r_{vi}^2}} \quad (1)$$

The variable sim_{uv} and r_u and r_v represent the degree of similarity between users u and v and the ratings vectors $\vec{r}_u$ and $\vec{r}_v$. A user's rating on item I is represented by r_{ui}, and the 2-norms of that user's rating are represented by $\|\vec{r}_v\|_2$, and the ratings of users u and v are represented by $\|\vec{r}_u\|_2$, respectively (Li et al., 2017). There are two sets of ratings here: I_u and I_v, which indicate the sets of objects for which both u and v provided ratings. In this scenario, I_u and I_v represent the sets of items that were given ratings by both u and v. Completing the following steps will result in the PCC (Li et al., 2017) and is calculated using Eq. (2).

$$sim_{uv} = \frac{\sum_{i\in I_{uv}}(r_{ui}-\bar{r_u})(r_{vi}-\bar{r_v})}{\sqrt{\sum_{i\in I_{uv}}(r_{ui}-\bar{r_u})}\sqrt{\sum_{i\in I_{uv}}(r_{vi}-\bar{r_v})}} \tag{2}$$

whereas $\bar{r}_u$ and $\bar{r}_v$ represent the average ratings from u and v.

2. **Find out the nearest neighbor**
Typically, one of two strategies is used to locate the nearest neighbor: k-nearest neighbor or setting a threshold for the number of neighbors. Users that have the most in common with the present user u are considered to be his or their closest neighbors if they are selected using the k-nearest neighbor method. Users v are picked as one of the nearest neighbor current users in circumstances where both active user u and also the target user v have a high degree of resemblance (De et al., 2020).
3. **Determining rating predictions**
It is the prediction of ratings and the display of a top-N suggestion list. These are the two most essential methods of providing recommendations to an active online user who is actively exploring the web (Manouselis et al., 2020). Following both, the ratings on a new item I from users who are most similar to the active user u must be predicted by both, based on the ratings on the new item I from users who are most similar to the active user u ratings on a new item I. To obtain the predicted ratings, the calculations given in Eq. (3) should be used.

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v\in N_u} sim_{uv}(r_{vi}-\bar{r}_v)}{\sum_{v\in N_u}|sim_{uv}|} \tag{3}$$

where N_u shows the collection of users who are the same as user u. Some of the following circumstances may benefit from top-N recommendations: In general, shopping portals and sites that do not publish user ratings are good examples of this type of web page design. The user may be interested in certain items, and helpful data may be collected so that a matrix of user items with each member having either 0 or 1 may be created from the user's feedback information (Mirjalili, 2016). Predicted ratings are listed in decreasing order, with the top-N items being recommended to users at the end of the modeling process, and the predicted ratings are then sorted in decreasing order. Using memory-based classification for binary data, in the feedback matrix R, if the pair(u, i) i.e user–item is recognized, $r_{ui} = 1$ and if pair(u, i) is not detected then $r_{ui} = 0$. This is why the cosine vector similarity is determined using binary ratings, as given in Eq. (4) (Moradi & Ahmadian, 2015).

$$sim_{uv} = cos(\vec{r}_v, \vec{r}_v) = \frac{\sum_{i\in I_{uv}} r_{ui}.r_{vi}}{\sqrt{\sum_{i\in I_u} r_{ui}^2}\sqrt{\sum_{i\in I_v} r_{vi}^2}} = \frac{|I_u \cap I_v|}{\sqrt{|I_u|}\sqrt{|I_v|}} \tag{4}$$

It is important to note that I_u and I_v show the collections of objects observed by users v and u.

2.1.2. Item-based collaborative filtering

CF recommendation system based on item follows the same three-step process as the user-based CF recommendation system method.

1. **Calculate item similarity using user–item rating**
In statistical analysis, the PCC and the cosine vector are two often used metrics of item-to-item correlation. Eq. (5) given below is used for the calculation of adjusted cosine vector method (De et al., 2020).

$$sim_{ij} = \frac{\sum_{u\in U_{ij}}(r_{ui}-\bar{r_u})(r_{uj}-\bar{r_u})}{\sqrt{\sum_{u\in U_i}(r_{ui}-\bar{r_u})^2}\sqrt{\sum_{u\in U_j}(r_{ui}-\bar{r_u})^2}} \tag{5}$$

A sim_{ij} symbol indicates that two objects i and j are similar in some way. Item I was rated by the user groups U_i and U_j, while item j was rated by the user group U_{ij}, which is the user group rated by both items i and j.
2. **Find out the Nearest Neighbor**
The user-based CF technique and the k-nearest neighbor strategy are frequently used to calculate the nearest neighbors in a CF system based on item strategies.
3. **Determine rating predictions**
To determine the rating prediction, the given Eq. (6) can be used (Qian et al., 2018).

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j\epsilon N_i} Sim_{ij}(r_{uj}-\bar{r}_j)}{\sum_{j\epsilon N_i}\left|sim_{ij}\right|} \tag{6}$$

where Ni represents the item i similar neighbors set.

2.2. Model-based collaborative filtering

In addition to being simple to use, memory-based recommendation systems are also simple to understand. While a memory-based recommendation system is useful for applications with a high number of users and items, it is not ideal for all applications, it is not suitable for applications with low user and item counts. As a result, the model-based recommendation system arrives later in this case, which may allow us to avoid some of the more important drawbacks (Gong et al., 2018), which are listed below. Model-based recommendation systems require a pre-learning stage before producing a recommendation to acquire the appropriate model parameters. A recommendation system based on a model may reliably predict user ratings in a relatively short amount of time once the learning phase is completed after the learning phase has been completed (Huang, 2019).

2.2.1. Matrix factorization model

A large proportion of matrix factorization (MF) models makes use of the latent feature model (LFM) (Wu et al., 2018). Even though some researchers have attempted to reduce the size of their databases, Matrix Factorization is the most successful option for dealing with the problem of high degrees of sparsity in the Research System database. It is common practice to apply the latent Semantic Index (LSI) and the Singular Value Decomposition (SVD) reduction method when employing a model-based approach (Koohi & Kiani, 2021). To begin with, we have a collection of U users and a collection of I-items. Let R be the size of the matrix $|U| \times |I|$ holding all of the user ratings for the items. By using this, the previously undiscovered characteristics will be discovered (Raza & Ding, 2019). Our goal here is to construct two matrices, one is$P(|U| \times K), Q(|I|)$, whose product is approximately comparable to the R provided by the original matrix (Raza & Ding, 2019) as shown in Eq. (7) given below.

$$R \approx PQ^T \tag{7}$$

By creating a map between people and objects, matrix factorization models can create a combined latent factor space with dimension f, where human interaction is represented as internal products (Chang et al., 2016). Thus, each object I am associated with a q_i Rf vector, and each user u is associated with an q_u Rf vector, as shown in the diagram below. I measure the good and negative aspects of q_i for a specific thing by determining the extent to which the item possesses those positive and negative factors. Consequently, the dot product $q_{iT} p_u$ that is produced encapsulates not just the interaction between user u and the item I, but also the overall interest in the item's attributes that the users have. In this case, the estimate is calculated based on the user u rating of item I, which is denoted by the variable r_{ui} (Madani et al., 2019) as in Eq. (8).

$$\hat{r}_{ui} = Qi^T P_u \tag{8}$$

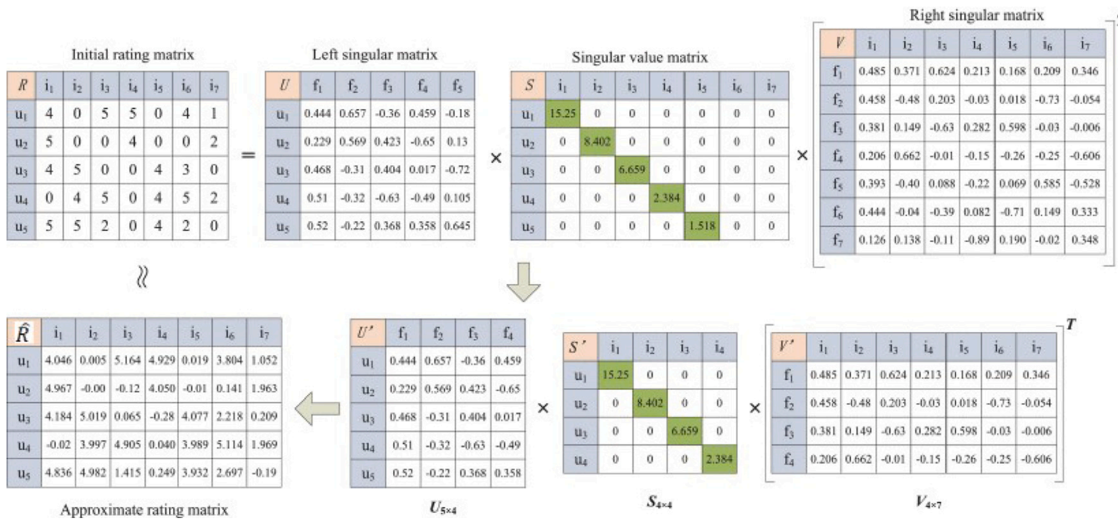

Fig. 4. An SVD matrix decomposition process (Huang, 2019).

2.2.2. Non-negative matrix factorization model

Non-negative matrix factorization (NMF or NNMF) and non-negative matrix measurement (NMM or NNMM) are the two procedures in different areas of analysis and line algebra in which an element of matrix V is (typically) embedded in two other elements of matrix W and H, with the advantage that none of the three matrices has any negative properties. Because there is no negative, it is simple to test the new matrices that are emerging (Zhoubao et al., 2015). There is also no negative in applications such as sound spectrogram processing or muscle function, which is compatible with the absence of negative data. Because the problem is rarely solved, it is typically quantified in terms of numbers. Among other things, NMF has applications in astronomy, computer vision, document compilation, missing data, chemometrics, audio signal processing, recommendation systems, and bioinformatics, to name a few fields (Zhang et al., 2020).

2.2.3. Singular value decomposition

A large amount of data sparsity is a major problem in recommendation systems. This has resulted in dimensionality reduction being identified as a serious issue that has to be handled immediately, and SVD, which is a specialization of the MF algorithms, has proven to be a highly effective tool for this purpose (Sarik & Mohammad, 2015). According to SVD technology, an initial rating matrix $_{m\times n}$ can be deconstructed into the elements U, S, and V as shown in Eq. (9) given below.

$$R_{m\times n} = U_{m\times m} S_{m\times n} V^T_{n\times n} \tag{9}$$

$U^T U = I_{m\times n}$, and $V^T V = I_{n\times n}$ in this example. Every column of U is referred to as a left singular vector, while each row of V^T is referred to as a right singular vector. The diagonal values are sorted in the order of the diagonal values from big to tiny where S is a diagonal matrix, which is referred to as single values. RR^T or $R^T R$ square root is used to indicate the diagonal value on the S matrix (Huang, 2019). The SVD matrix decomposition approach is shown in Fig. 4.

The initial matrix R dimension is reduced, as indicated by the letters U, S, and V in Fig. 4. Where U represents the information about the user, V represents the information about an item, and S shows the feature relevance. The first four qualities were chosen since they account for over 96 percent of total energy (Huang, 2019). In terms of size, R approaches the true matrices R.

2.3. Challenges in collaborative filtering recommendation system

When the volume of data increases, then the data type becomes richer and it became more complicated for the application environment. The following significant issues are caused by the following factors.

2.3.1. Data sparsity

In the user object matrix, there are numerous unknown ratings, and sparsity frequently occurs more and more. Excessive smallness leads to a large number of comparable ratings among very few or no items, as well as a substantial difference in the number of similarities, which affects the recommendation's quality. As a result, a good recommendation system should consider the amount of data available (Gong et al., 2018).

2.3.2. Interpretability

The interpretation of a CF-based recommendation system is one of the few issues that they have to deal with. Quality of algorithms cannot be established just by utilizing algorithms quality tests such as MAE. The act of recommending goods to users who place a high emphasis on accuracy simply wastes resources and delivers marginal benefit. They will be unable to determine whether or not the recommended items meet the demands of the users if they are unable to adequately define the desired results, resulting in decreased system reliability as a result. The recommendation system's ability to supply some explanatory information along with its recommendations may significantly improve the credibility of the proposed outcomes (Chen et al., 2018).

2.3.3. Cold start

The system does not know anything about a new user or item when it first encounters them because it does not know the user's history or the item's user ratings. As a result, the system cannot provide the user with a recommendation service and the user is presented with an item that is difficult to recommend to the system. It is typical to employ mixed-use suggestions to solve this problem (such as user age, user interaction relationships, product tags, etc. Mixed-use ideas integrate ratings and information content) (Dhruv et al., 2019).

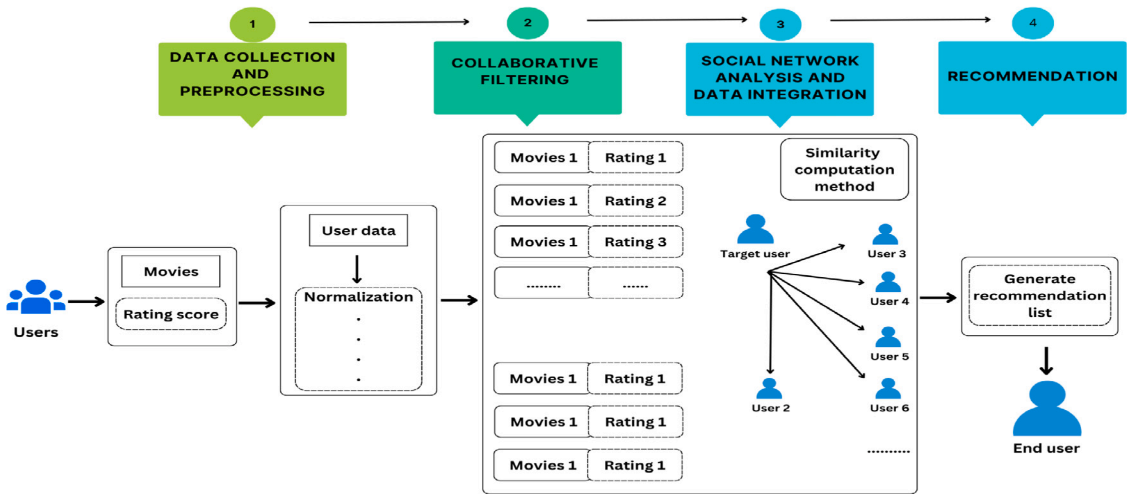

Fig. 5. Framework of collaborative filtering recommendation model.

3. Proposed model

The proposed model combines the strengths of CF and social network analysis to provide more accurate and relevant recommendations to users as shown in Fig. 5. The system collects both user–item interaction data and social network data and performs social network analysis to identify the connections between users and compute various measures such as the strength of connections, common interests, and influence. CF methods such as matrix factorization are then used to compute user–item preferences from the interaction data and generate a user–item preference matrix. The social network data is used to weigh the user–item preferences, and the weighted preferences are used to generate recommendations for users. By integrating social network data into the CF recommendation system, the proposed model aims to capture the complex relationships between users and items and provide more personalized recommendations.

3.1. Data collection and preprocessing

In this step, we collected data from various sources and cleaned, formatted, and prepared it for analysis. This data includes user behavior data such as ratings, reviews, and purchase history, as well as social network data such as user connections and interactions.

It is possible to get misleading results when analyzing data that has not been carefully reviewed for the aforementioned concerns. The major data are processed while we are in this phase to fulfill the demand of the recommendation approach before commencing the computation of similarity. We standardize user ratings to guarantee that our proposed approach for creating recommendations can directly compute this data. Because the appearance of certain implicit assessments does not always represent users' preferences, but it is also hard to ignore them, we assign them suitable values.

As an example, while working on real data, Table 1 shows the preprocessed Movielens dataset after applying the above preprocessing steps in Algorithm 1. Similarly, Table 2 shows the preprocessed Bookcrossing dataset.

[1] MovieLens dataset: https://grouplens.org/datasets/movielens/
[2] Book-Crossing dataset: http://www2.informatik.uni-freiburg.de/cziegler/BX/

Algorithm 1 Data Preprocessing for Movielens and BookCrossing Datasets

Require: User-item interaction data R, social network data S, dataset type (Movielens or BookCrossing)
Ensure: User-item interaction data R', social network data S'
1: Remove duplicate entries from R and S
2: Remove users and items with less than k interactions, where k is a hyperparameter (set to 5 for Movielens and 10 for BookCrossing)
3: **if** dataset type is Movielens[1] **then**
4: Remove users and items with extreme ratings (e.g., ratings below 1 or above 5 for Movielens)
5: **else if** dataset type is BookCrossing[2] **then**
6: Remove users and items with implicit feedback (i.e., ratings not in the range [1, 10] for BookCrossing)
7: **end if**
8: Normalize the rating values to have zero mean and unit variance
9: Split R' into training, validation, and test sets, with ratios of 70
10: Create the social network adjacency matrix A from S'
11: Convert A to a binary matrix by setting all non-zero entries to 1
12: Split A into training, validation, and test sets, with the same ratios as R'
13: **return** R', S', and A

Table 1
Preprocessed movielens dataset.

User ID	Movie ID	Rating
1	2	0.507
2	1	0.712
3	3	1.414
4	4	0.326
5	6	1.617
6	8	0.235
7	7	0.000

3.2. Integration of social network analysis and data

In this step, the network visualization technique is used to analyze social network data to identify influential users and communities within the network, then we combined the results from the CF recommendation system and social network analysis to enhance the accuracy and relevancy of recommendations. The influential users or communities

Table 2
Preprocessed book-crossing dataset.

User-ID	ISBN	Book-Rating
276725	034545104X	3
276726	0155061224	1
276727	0446520802	2
276729	052165615X	7
276729	0521795028	9
276733	2080674722	6
276736	3257224281	4
276737	0600570967	8
276744	038550120X	5
276745	342310538	10

identified through social network analysis may be given greater weight in the recommendation algorithm.

3.2.1. Similarity computation

In the proposed framework, the similarity computation method is a weighted combination of two components: user–item ratings and social connections. The adjacency matrix is given below;

$$A = \begin{bmatrix} sim(u_1,u_1) & sim(u_1,u_2) & \cdots & sim(u_1,u_n) \\ sim(u_2,u_1) & sim(u_2,u_2) & \cdots & sim(u_2,u_n) \\ \vdots & \vdots & \ddots & \vdots \\ sim(u_n,u_1) & sim(u_n,u_2) & \cdots & sim(u_n,u_n) \end{bmatrix}$$

To perform similarity computation on the constructed adjacency matrix, we will use cosine similarity as our similarity metric.

A be the constructed adjacency matrix for a dataset, where $A_{i,j}$ represents the connection between user i and user j. Then, the cosine similarity between users i and j can be computed as follows:

$$\text{similarity}(i,j) = \frac{\sum_{k=1}^{n} A_{i,k} \cdot A_{j,k}}{\sqrt{\sum_{k=1}^{n} A_{i,k}^2} \cdot \sqrt{\sum_{k=1}^{n} A_{j,k}^2}}$$

where n is the total number of users.

Using this formula, we can compute the cosine similarity between all pairs of users in the constructed adjacency matrix for both the Movielens and BookCrossing datasets.

3.3. Recommendation

The target user's similarity and other users will be calculated using the effect of the target user in this phase. For each product the target buyer did not choose as their first choice, we will now calculate their overall preference degree. Finally, the items with the highest overall preference degree are those that are utilized to build a suggestion list that is organized in a diminishing fashion. Finally, the proposed model can provide suggestions for products in the L category's top tier to our target audience.

4. Results

In this section, all the experimental results of the proposed model of CF based on social networks are discussed, and also the comparison with other models is discussed in detail. By using two standard datasets, it has been tested how well the recommendation system performed. The results were quite encouraging. The MovieLens dataset in Moradi and Ahmadian (2015) has a total of 1584 films as well as 845 actors and actresses. Every user has rated at least 21 movies on a scale that goes from 1 to 5, giving each one a rating between 1 and 5 based on their general opinion of the movie. A total of one hundred thousand ratings are included in the initial data set. This collection is made up of a total of three distinct kinds of information tables. In addition, a rating has been assigned to each movie. As part of our study, the datasets will need some preliminary processing. Only links with a rating of at least three

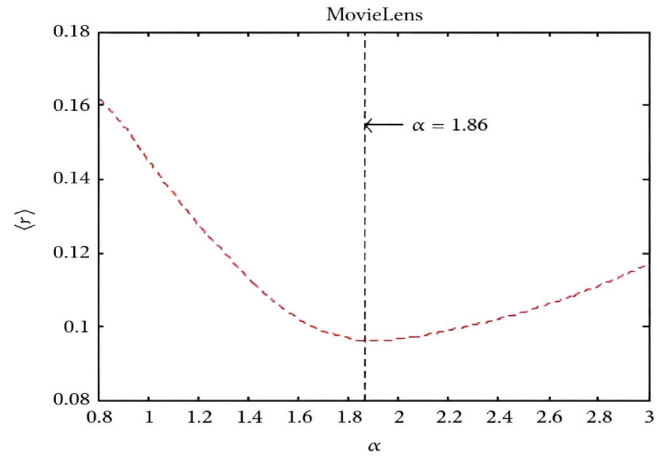

Fig. 6. Resultant graph of Movielens dataset.

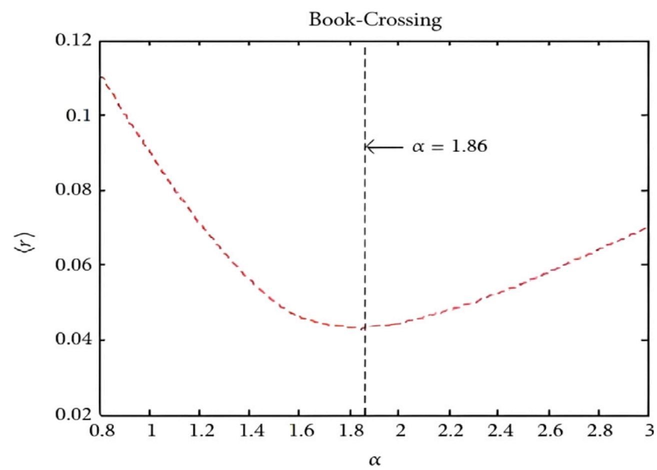

Fig. 7. Resultant graph of Book-Crossing dataset.

stars are considered by MovieLens, with VLI equating to the numerals three, four, and five.

The Book-Crossing dataset (Qian et al., 2018) is comprised of 277,848 individuals, all of whom have had their names obscured but have provided demographic information. These individuals are considered as a group, given a total of 1,049,770 stars based on their opinions about 261,369 different books. Ratings (Books) can either be explicit, in which case they are stated as a discrete number on a scale ranging from 1 to 10 (with higher values indicating a higher level of appreciation), or implicit, in which case they are expressed as 0. In the case of explicit ratings, the scale whose range is 1,2,3...10, and the higher values indicate appreciation of a higher level.

Only links with a rating of 0 or more than 5 are considered for Book-Crossing, and the value for VLI is "10 to 1". Every processed dataset is split into two sets of data, the training set, which is composed of 80 percent of the entire data, and the test set, which is composed of 20 percent of the remaining data. Initially, we predicted the range of ideal parameter values to limit the amount of calculation required for the proposed approach by obtaining the best parameter values. Based on several published research on CF techniques, the values that have produced the best results vary between 1 to 2. Repeated computations using a binary search method has done to quickly find the best parameter values. A difference of 0.1 is selected throughout

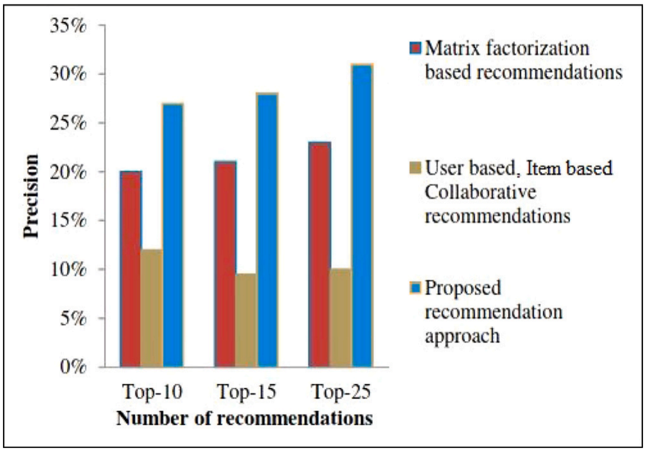

Fig. 8. Precise comparison between the proposed approach and other recommendation approaches.

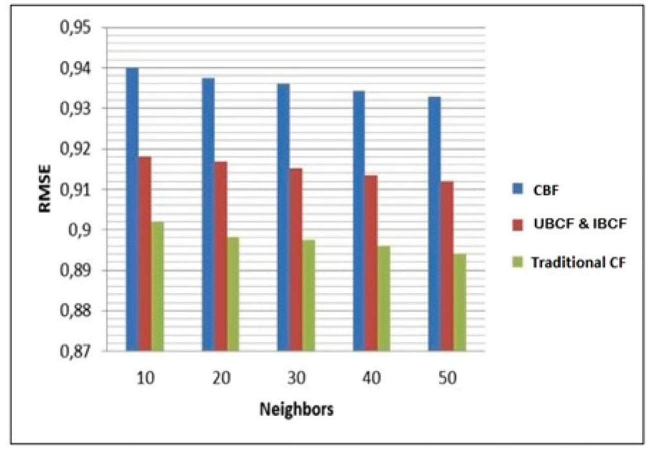

Fig. 9. RMSE comparison of Traditional CF, UBCF, IBCF, and CBF.

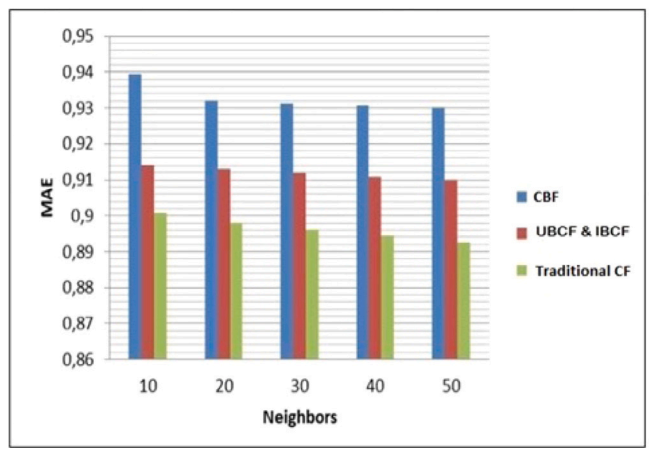

Fig. 10. MAE comparison of Traditional CF, UBCF, IBCF, and CBF.

the iterative calculation. Computing expenses will be decreased if all of these computational definitions and processes are used. It has been noticed that the optimum performance of this strategy occurs at a value of roughly 1.86 as can be seen in Figs. 6 and 7 for both test datasets. Individuals may, in reality, change the value of the parameter.

4.1. Performance measure comparison with other models

Searching for target users, and similar users are crucial for the majority of the CF techniques as it is considered that similar users in a social network prefer similar products. In comparison to these (Sarik & Mohammad, 2015; Zhoubao et al., 2015) models, CF techniques utilize product reviews from customers to find other people who had those opinions. Using only user ratings of products is insufficient for recommending desired items and searching for similar users. To update the user rating of things for recommendations utilizing similar users, our proposed model first employs social information merging to look for users who are similar to them.

4.1.1. Precision

The majority of the users do not like to view lengthy recommendation lists. Consequently, the system is assessed using a precision assessment measure to see how well it performs on a shorter recommendation list. The precision is defined in Eq. (10).

$$Precision = \frac{\text{Total number of correct recommendations}}{\text{Total number of recommendations generated}} \tag{10}$$

The proposed approach is then compared to other benchmark recommendation systems such as user-based CF as well as the matrix factorization method. Fig. 8 illustrates the precision comparisons graph of our proposed approach with various standard approaches.

For smaller recommendation lists, the existing benchmark model's precision value is quite low. Compared to the traditional user-based CF technique and the most recent matrix factorization recommendation technique for the top-25 suggestions, the proposed approach has significantly improved by 21 percent and 8 percent, respectively. By including users' context-related data in the recommendation system, the method can be further improved.

4.1.2. Root mean square error

Root Mean Square Error (RMSE) is the metric most frequently used to measure how recommendation systems perform over time. As a result, we used those criteria to assess how well the traditional CF, user-based collaborative filtering (UBCF), item-based collaborative filtering (IBCF), and content-based filtering (CBF) algorithms performed when making recommendations.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} p(i) - q(i)^2}{N}} \tag{11}$$

where $p(i)$ shows the predicted value and $q(i)$ shows the actual value while N represents the total number of observations.

The RMSE metrics findings for the proposed strategies are presented in Fig. 9. The graph shows that, in comparison to the Proposed CF method, the RMSE results of the UBCF, IBCF, and CBF are high. Thus the CF method has obtained better result than the other two algorithms

4.1.3. Mean absolute error

We used the same criteria for mean absolute error (MAE) as used in RMSE to assess how well the traditional collaborative filtering (CF), user-based collaborative filtering (UBCF), item-based collaborative filtering (IBCF), and content-based filtering (CBF) algorithms performed when making recommendations. See Eq. (12) for the MAE findings w.r.t neighbors.

$$MAE = \frac{\sum_{i=1}^{N} \left| p(i) - q(i)^2 \right|}{N} \tag{12}$$

As shown in Fig. 10, the MAE results of the traditional CF technique are equally low across the whole neighbor range when compared to the UBCF, IBCF, and CBF. The CF-based method obtains higher accuracy by achieving low MAE value than the two other algorithms.

Based on Figs. 9 and 10, in terms of RMSE and MAE, we can see that the conventional CF algorithms surpass the classic algorithm of CBF. In addition, integrating the CF approach with the K-means clustering technique has considerably improved the performance of recommendations, demonstrating that the CF-based method is a better algorithm to be utilized for big data recommendation systems.

4.2. Discussion

The proposed research presents a novel collaborative filtering (CF) recommendation framework that utilizes social networks to overcome the sparsity issue that traditional CF methods often face. Compared to other feature-based CF methods, our approach incorporates social connections between users to introduce diversity in recommended items and achieves this through a weighted combination of user ratings and social connections, which are learned through an optimization process. By incorporating the preferences of a user's friends, we aimed to better predict a user's preferences for items that they have not yet interacted with. Our experimental results on a real-world dataset demonstrate that our proposed method outperforms several state-of-the-art CF and feature-based CF methods in terms of recommendation accuracy, diversity, and relevance. Thus, our proposed CF method with feature fusion provides a unique and innovative approach to addressing the limitations of traditional CF and feature-based CF methods, and our experimental results show its potential advantages in recommendation tasks.

Our research makes several contributions to the field of recommendation systems. Firstly, it provides a novel approach to addressing the sparsity issue in CF-based recommendation frameworks by incorporating social network information. Secondly, it demonstrates that this approach leads to more accurate and diverse recommendations for users. Finally, it indicates that the recommendations generated by our framework are well-perceived by users. Additionally, the proposed framework could be extended to incorporate other forms of side information, such as demographic information or item content, to further improve the performance of the recommendations. Overall, our proposed recommendation framework utilizing social networks is a valuable tool for recommendation systems. It effectively addresses the sparsity issue of traditional CF methods and improves the accuracy and diversity of recommendations.

5. Conclusion and future work

The framework takes into account both user–item interactions and social connections between users to make recommendations. The proposed framework outperforms traditional CF techniques in terms of recommendation accuracy. This is demonstrated through experiments conducted on real-world datasets, where the proposed framework shows an improvement in recommendation performance as measured by several evaluation metrics, such as precision and recall. Additionally, incorporating social network information into the recommendation process provides valuable information about users and can help to mitigate the sparsity problem in traditional CF techniques. The results also show that the proposed framework is scalable and can handle large datasets with millions of users and items. Overall, the proposed framework effectively leverages social network information to improve recommendation accuracy and provides a promising direction for future work in recommendation systems.

The proposed framework effectively leverages social network information to improve recommendation accuracy and provides a promising direction for future work in recommendation systems. However, there are several areas where the framework could be further improved. One potential avenue for future research is to incorporate more sophisticated methods for integrating social network information into the recommendation process. For example, this could involve exploring different ways of representing social network structures or incorporating additional forms of contextual information. Additionally, exploring real-time recommendation methods and privacy protection techniques would help to address scalability and privacy concerns, respectively (Wu et al., 2018).

Another potential area for future research is to explore the effectiveness of the proposed framework in other domains beyond the movie recommendation system used in this study. For example, the framework could be applied to e-commerce or social media platforms to make more personalized recommendations for products or content. Finally, future work could also involve evaluating the proposed framework in a real-world setting to assess its practicality and effectiveness. By addressing these limitations and further exploring these research directions, the proposed framework could be enhanced to provide even more accurate and relevant recommendations.

CRediT authorship contribution statement

Aamir Fareed: Investigation, Methodology, Writing – original draft. **Saima Hassan:** Supervision, Writing – original draft. **Samir Brahim Belhaouari:** Formal analysis, Writing – review & editing. **Zahid Halim:** Formal analysis, Writing – review & editing.

Declaration of competing interest

There is no conflict of interest, or disclose all the conflicts of interest regarding the submitted manuscript.

Data availability

Link of data is provided in the manuscript.

Acknowledgment

Open Access funding provided by the Qatar National Library.

References

Chang, X., Yu, Y.-L., Yang, Y., & Xing, E. P. (2016). Semantic pooling for complex event analysis in untrimmed videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(8), 1617–1632.

Chen, R., Hua, Q., Chang, Y.-S., Wang, B., Zhang, L., & Kong, X. (2018). A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. *IEEE Access*, *6*, 64301–64320.

De, M. C., Limongelli, C., Sciarrone, F., & Temperini, M. (2020). MoodleREC: A recommendation system for creating courses using the moodle e-learning platform. *Computers in Human Behavior*, *104*, Article 106168.

Dhruv, A., Kamath, A., Powar, A., & Gaikwad, K. (2019). Artist recommendation system using hybrid method: A novel approach. In *Emerging research in computing, information, communication and applications* (pp. 527–542). Springer.

Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., & Yin, D. (2019). Graph neural networks for social recommendation. In *The world wide web conference* (pp. 417–426).

Fessahaye, F., Perez, L., Zhan, T., Zhang, R., Fossier, C., Markarian, R., Chiu, C., Zhan, J., Gewali, L., & Oh, P. (2019). T-recsys: A novel music recommendation system using deep learning. In *2019 IEEE international conference on consumer electronics* (pp. 1–6). IEEE.

Gong, C., Chang, X., Fang, M., & Yang, J. (2018). Teaching semi-supervised classifier via generalized distillation. In *IJCAI* (pp. 2156–2162).

Huang, C.-L. (2019). Bayesian recommender system for social information sharing: Incorporating tag-based personalized interest and social relationships. *Intelligent Data Analysis*, *23*(3), 623–639.

Koohi, H., & Kiani, K. (2021). A new method to find neighbor users that improves the performance of collaborative filtering. *Expert Systems with Applications*, *83*, 30–39.

Li, Y., Wang, D., He, H., Jiao, L., & Xue, Y. (2017). Mining intrinsic information by matrix factorization-based approaches for collaborative filtering in recommender systems. *Neurocomputing*, *249*, 48–63.

Li, C., Wang, Y., Tang, H., & Luo, Y. (2019). Dynamic multi-objective optimized replica placement and migration strategies for SaaS applications in edge cloud. *Future Generation Computer Systems*, *100*, 921–937.

Madani, Y., Erritali, M., Bengourram, J., & Sailhan, F. (2019). Social collaborative filtering approach for recommending courses in an E-learning platform. *Procedia Computer Science*, *151*, 1164–1169.

Manouselis, N., Drachsler, H., Verbert, K., & Santos, O. C. (2020). Workshop on recommender systems for technology enhanced learning. In *Proceedings of the fourth ACM conference on recommender systems* (pp. 377–378).

Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, *96*, 120–133.

Moradi, P., & Ahmadian, S. (2015). A reliability-based recommendation method to improve trust-aware recommender systems. *Expert Systems with Applications*, *42*(21), 7386–7398.

Naseri, S., Bahrehmand, A., & Ding, C. (2015). An improved collaborative recommendation system by integration of social tagging data. In *Recommendation and search in social networks* (pp. 119–138). Springer.

Paradarami, T. K., Bastian, N. D., & Wightman, J. L. (2017). A hybrid recommender system using artificial neural networks. *Expert Systems with Applications*, *83*, 300–313.

Priambodo, A., Handayani, P. W., & Pinem, A. A. (2019). Success factor for IT project implementation in banking industry: A case study. In *2019 3rd international conference on informatics and computational sciences* (pp. 1–5). IEEE.

Qian, X., Feng, H., Zhao, G., & Tao, M. (2018). Personalized recommendation combining user interest and social circle. *IEEE Transactions on Knowledge and Data Engineering*, *26*(7), 1763–1777.

Rajesh, S., Paul, V., Menon, V. G., & Khosravi, M. R. (2019). A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices. *Symmetry*, *11*(2), 293.

Raza, S., & Ding, C. (2019). Progress in context-aware recommender systems—An overview. *Computer Science Review*, *31*, 84–97.

Sarik, G., & Mohammad, A. N. (2015). Enhancing memory-based collaborative filtering for group recommender systems. *Expert Systems with Applications*, *42*(7), 3801–3812, URL https://www.sciencedirect.com/science/article/pii/S0957417414007386.

Shahbazi, Z., Hazra, D., Park, S., & Byun, Y. C. (2020). Toward improving the prediction accuracy of product recommendation system using extreme gradient boosting and encoding approaches. *Symmetry*, *12*(9), 1566.

Shen, J., Zhou, T., & Chen, L. (2020). Collaborative filtering-based recommendation system for big data. *International Journal of Computational Science and Engineering*, *21*(2), 219–225.

Shrestha, Y. R., & Yang, Y. (2019). Fairness in algorithmic decision-making: Applications in multi-winner voting, machine learning, and recommender systems. *Algorithms*, *12*(9), 199.

Tahmasebi, H., Ravanmehr, R., & Mohamadrezaei, R. (2021). Social movie recommender system based on deep autoencoder network using Twitter data. *Neural Computing and Applications*, *33*(5), 1607–1623.

Wu, H., Zhang, Z., Yue, K., Zhang, B., He, J., & Sun, L. (2018). Dual-regularized matrix factorization with deep neural networks for recommender systems. *Knowledge-Based Systems*, *145*, 46–58.

Yao, L., Sheng, Q. Z., Qin, Y., Wang, X., Shemshadi, A., & He, Q. (2018). Context-aware point-of-interest recommendation using tensor factorization with social regularization. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval* (pp. 1007–1010).

Zeng, Z., Yiming, S., Qinfen, L., Bocheng, W., David, G., & Christopher, G. (2017). Investigation of novel partitioned-primary hybrid-excited flux-switching linear machines. *IEEE Transactions on Industrial Electronics*, *65*(12), 9804–9813.

Zhang, Y., Feng, F., He, X., Wei, T., Song, C., Ling, G., & Zhang, Y. (2021). Causal intervention for leveraging popularity bias in recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 11–20).

Zhang, F., Qi, S., Liu, Q., Mao, M., & Zeng, A. (2020). Alleviating the data sparsity problem of recommender systems by clustering nodes in bipartite networks. *Expert Systems with Applications*, *149*, Article 113346.

Zhoubao, S., Lixin, H., Wenliang, H., Xueting, W., Xiaoqin, Z., Min, W., & Hong, Y. (2015). Recommender systems based on social networks. *Journal of Systems and Software*, *99*, 109–119.