A Hybrid Recommender System Based-on Link Prediction for Movie Baskets Analysis

Mohammadsadegh Vahidi Farashah\textsuperscript{a}, Akbar Etebarian\textsuperscript{b*}, Reza Azmi\textsuperscript{c}, Reza Ebrahimzadeh Dastjerdi\textsuperscript{d}

\textsuperscript{a}Mohammadsadegh Vahidi Farashah
Ph.D. Candidate of Information Technology Management, Department of Management, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran.
sadegh.vahidi@khuisf.ac.ir, +989125055404

\textsuperscript{b*}Akbar Etebarian
Associate Professor of Management, Department of Management, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran. Corresponding Author: etebarian@khuisf.ac.ir, +989132105784

\textsuperscript{c}Reza Azmi
Associate Professor of Computer Engineering, Department of Engineering and Technology, Alzahra University, Tehran, Iran.
azmi@alzahra.ac.ir, +989121547174

\textsuperscript{d}Reza Ebrahimzadeh Dastjerdi
Assistant professor of Management, Department of Management, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran.
Ebrahimzadeh@gmail.com, +989132252272
A Hybrid Recommender System Based-on Link Prediction for Movie Baskets Analysis

Abstract
The proposed method in this paper consists of three steps: initial clustering of all users and assigning new user to appropriate clusters, assigning appropriate weights to users' characteristics, and identifying new user's adjacent users using hybrid similarity criteria and adjacency matrix of adjacent users’ rating to the movie services and calculating new user’s rating to each movie considering adjacent users’ rating and the similarity level of each adjacent user to the new user. The results show that the mean squared error of the proposed model has decreased respectively 8.59%, 8.67%, 8.45% and 8.15% compared to the basic models such as Naive Bayes, multi-attribute decision tree and randomized algorithm. Also, MAE of the proposed method decreased by 4.5% compared to SVD and approximately 4.4% compared to ApproSVD and RMSE of the proposed method decreased by 6.05% compared to SVD and approximately 6.02% compared to ApproSVD.

Keyword: hybrid recommender system, cold start problem, link prediction, hybrid similarity criterion

Introduction
Due to the high importance of recommender systems in social networks, real life, e-commerce, shopping cart analysis, etc., a lot of research has been done in recent years [1–3]. Recommender systems are one of the most popular systems that have attracted the attention of various researchers during the past decade. Recommender systems are used to filter huge amount information, such as users’ cart [4]. Recommender systems are used in a variety of fields such as shops, libraries, restaurants, tourism systems, shopping carts and other environments to provide attractive items such as movie services [5]. These systems play an important role in e-commerce [6]. Due to the huge amount of information that exists,
providing the most appealing services with high accuracy and appropriate time is one of the important issues. The service recommender system enables users to review products having features such as product’s name, manufacturer, production date, brand type, and so on. For users who are new and there is not enough information about them in the system (they have cold start problem), the recommender system offers a list of products which are rated by other users [7]. One of the most important challenges of recommender systems is the challenge of user’s cold start [8]. The problem of cold start occurs when the user has no activity or transaction in the system. Due to the cold start problem of users, a variety of recommender systems have been proposed. In general, recommender systems are divided into two categories:

- **Traditional Recommender Systems**
  - A) **Content-Base Filtering** [9]
  - B) **Collaborative Filtering** [10]
  - C) **Hybrid Recommender Systems** [11]

- **Modern Recommender Systems**
  - A) **Demographic-Base Approach** [12]
  - B) **Knowledge-based Approach** [13]

The methods that have been studied by various researchers are collaborative and content-based filtering systems. Content-based systems classify users based on their demographic information. Collaborative filtering systems are one of the most widely used recommendation techniques that offer users the items that have been rated or selected by other similar users [14]. For example, if two users have similar interests and behaviors, they recommend the purchased service system (film) to each other [15]. In this system, unlike content-based systems, similar users are identified and items which are highly rated are offered to them. This method is used to present a list of products to a group of users using data mining
(clustering) techniques [16]. Using similarity criteria in collaborative systems to find adjacent users or similar activities is one of the main requirements of making recommendations. Similarity criteria in recommender systems make it possible to identify similar users or services based on their demographic activity, category and information. In this study, similarity criteria were used in collaborative recommender system to offer the similarity level of items that are rated by other users to the new user in different steps.

In general, this paper presents a hybrid system based on content-based and collaborative recommender system for analyzing user's cart in an online movie system. In the content-based recommender system, DBScan clustering algorithm and deep neural network algorithm are used to determine basic categories for users based on demographic information and also to classify new users. One of the most important reasons for using DBScan algorithm for the initial clustering of users based on demographic information is its speed and the ability to support large amounts of information compared to other clustering algorithms. Also, the most important reason for using deep neural network algorithm to classify new users is its ability to support huge amount of information and hidden layers compared to other methods is classification. The deep neural network enables new users to be transferred to the target group with high accuracy. The collaborative recommender system uses a combination of similarity criteria and the improved FriendLink link algorithm to determine the similarity between new users and other users. With the hybrid similarity criteria, the similarity level of users and the new user is calculated in terms of a threshold. The improved FriendLink link algorithm is used to provide friend recommendations based on user communication in online movie system. The remainder of this paper will be presented as follows: Section 2 reviews the literature, Section 3 describes the proposed approach and architecture, and in Sections 4 and 5, the results are presented and the conclusions are discussed.
Review of the Literature

In this study, for making recommendations in movie systems, several researchers tried to solve the problem of cold start. Hung et al, (2011) mentioned the cold start problem concerning movies and users. They introduced an important traditional system of collaborative filtering. In this model, two matrixes of similarity were used, one of which showed the similarity between users and movies and the other one showed the similarity between users themselves. Then, concerning the mechanism of the discussed forecast, they made some recommendations to the users. One of the weaknesses of this study was the high memory usage concerning members (users) and movies which was due to the construction of several similarity matrixes [17]. Bobadilla et al, (2012) used the neural network as an RS of the collaborative filtering to reduce cold start issues for new users. They assessed the Movielens dataset and Netflix and due to the usage of non-numeric data, they used Jaccard Similarity Index [18]. Henc (2013) recommended movies to users by clustering movies and using k-means algorithm. He carried out it based on users’ comments about movies. Henc studied famous Movielens dataset and implemented the presentation for data collection with 10109 movies that were assessed by 2113 users [19]. Kamvatsus et al, (2014) introduced a model in which classification algorithms such as Naïvebays, decision tree, and random classification algorithm were used as similarity metrics in order to recommend movies to users. Also, they evaluated movielens dataset [20]. In order to enhance the performance of the system and to solve cold start problem, Luize et al (2015) posed the hybrid method including both collaborative filtering and demographic information. In this study, they used the hybrid co-clustering algorithm and knowing the machine for solving the cold start problem and evaluated Movielens, Jester, and Netflix dataset [21]. Sperli et al, in [22], provided a recommendation system to improve social networking approach. In this paper, an RS which is designed for big data applications is used to provide useful recommendations on
online social networks. The proposed technique is a collaborative and user-centric approach that exploits the interactions between users and creates multimedia content on one or more social networks in a new and effective way. Experiments on the data collected from several online social networks revealed the feasibility of the approach regarding the problem of social media proposition. Kutty et al, in [23], presented recommender systems for large social networks: reviewing challenges and solutions. This article states that social networks are crucial for networking, communication, and content sharing. Social networking applications generate a great deal of information on a daily basis, and social networks are subject to extensive research due to the heterogeneity of data and the structures within them, their size and dynamics. When such a large amount of data is used by recommender systems, the connection result can help to solve social business issues and to improve friends' recommendations. This article is a review article that has compared some trends with each other. Lin et al., in [24], developed a recommendation system based on neural network for recommending movies to users. Due to unimportant challenges like scalability, dispersion and user's confidence compared with cold start and movies which have been researched till now, the challenges have also been resolved with preprocessing, clustering and classification. Walek et al, (2020) in [25] the main objective of this article to propose a hybrid recommender system predictor for recommending suitable movies. This system contains a recommender module combining a collaborative filtering system, a content-based system, and a fuzzy expert system.

Table 1 summarizes previous approaches to movie recommendation and the cold start challenge. This table outlines the advantages and disadvantages of each method.
### Table 1. Summarizes previous approaches to movie recommendation and the challenge of cold start users

| Authors                                      | Ref  | Method                                                                 | Advantages               | Disadvantages               |
|----------------------------------------------|------|------------------------------------------------------------------------|--------------------------|-----------------------------|
| Kim, H. N., El-Saddik, A., & Jo, G. S        | [17] | Collaborative error-reflected models for cold-start recommendation system | 1. High speed            | 1. Low accuracy<br>2. Low precision |
| Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. | [18] | A collaborative filtering approach to mitigate the new user cold start problem | 1. Normal accuracy<br>1. Complex model |                             |
| Byström, H.                                  | [19] | Movie recommendations from user ratings                                 | 1. good accuracy<br>1. Complex model |                             |
| Lika, B., Kolomvatsos, K., & Hadjieffthymiades, S | [20] | Facing the cold start problem in recommender system<br>Simultaneous co-clustering and learning to address the cold start problem in recommendation system | 1. Fast Execution time<br>1. High MAE<br>2. High RMSE |                             |
| Pereira, A. L. V., & Hruschka, E. R.         | [21] | A social media recommendation system                                   | 1. High speed<br>1. Low accuracy<br>2. Low precision |                             |
| Sperlì, G., Amato, F., Mercorio, F., Mezzanzanica, M., Moscato, V., & Picariello, A. | [22] |                                                                           | 1. Normal accuracy<br>1. Complex model |                             |
| Kutty, S., Chen, L., & Nayak, R               | [23] | A people-to-people recommendation system using tensor space models       | 1. High speed<br>1. Low accuracy<br>2. Low precision |                             |
| Lin, C. H., & Chi, H                         | [24] | Novel Movie recommendation system based on collaborative filtering and neural networks | 1. Fast Execution time<br>1. High MAE<br>2. High RMSE |                             |
| B. Walek, V.Fojtik                           | [25] | Module combining a collaborative filtering system, and a content-based system, and a fuzzy expert system. | 1. High speed<br>1. Complex model |                             |

Figures (1) show the results of researches conducted from 2000 to 2019 in link prediction and recommender system.
As can be seen, a large number of research on link prediction were done in 2013 and 2014. The results of another study concerning the research which were conducted from 2000 to 2019 on recommender systems in social networks are shown in Fig. 2.

As can be seen in the figure above, a large number of research on recommender systems were carried out in 2013 and 2014. Also, in 2015 and 2016 a great number of studies were conducted. Figure 3 shows a comparison of the conducted researches from 2000 to 2019 in the field of link prediction and recommender systems.
The number of link prediction studies is about 0.5 times higher than that of recommender systems. Therefore, through reviewing the literature it has been observed that various methods have been proposed to provide users with recommendations in movie services. Each of these methods has its own challenges, such as inadequate accuracy, high error rate and lack of appealing services. This article presents a recommender system based on a combination of content-based and collaborative recommender systems which solves both the problem of cold start and addresses the challenge of users' trust.

The Proposed Method

In this section, we will describe and present the proposed method with regard to the flowchart that is being presented and in the following sections, the items specified in the flowchart will be described in full detail. As can be seen in Fig. 4, the Movielens dataset is first introduced into the hybrid recommender system. This database has three sections: datasets of user communications, demographic information and rated movies. The datasets of demographic information and rated movies are used in content-based recommender systems
and the user communication dataset is used in collaborative systems. After entering the dataset into the proposed system, the first phase of the proposed method is executed. The first phase involves content-based recommendation system. In content-based recommender system, DBScan clustering algorithm is used for the initial clustering of users’ dataset based on demographic information and deep neural network algorithm to classify new users. In this phase, all users are first clustered based on demographic information such as age, occupation and gender using DBScan algorithm.

![Block diagram of the proposed method](image)

**Figure 4. Block diagram of the proposed method**
For all users in the system, a label is specified as a cluster. Then the categories specified for each user are determined as label attributes. In phase 1, the content-based recommender system initializes the clustering and assigning new users to categories. After determining the categories, phase 2 of the proposed method begins. The phase includes a collaborative recommender system based on new similarity criteria. In this phase, similarities are calculated based on a threshold (lambda) between the new user and the users in the selected category. Similarity criteria are computed based on age, gender, and occupation. The collaborative recommender system extracts users who are the most similar to the new user. Then higher rated movie services are suggested to the new user based on the proximity matrix. After determining the movie services for the new user, phase 3 which is the Link Prediction Algorithm begins. At this stage, movies are transmitted through user communication dataset. In this phase, movies that are similar based on the communication between associated users and the new user are selected. Phase 4 of the proposed method is related to the output of collaborative recommender system and the improved FriendLink algorithm. Based on the block diagram above, movie services are offered to new users. Finally, the results are evaluated. Based on the phases described in the proposed method, the steps of the proposed method are as follows:

**Phase 1: Content-Based Recommender System**

After loading the dataset into the proposed system, the first phase of the proposed method is executed. The first phase involves a content-based recommender system. In the content-based recommender system, DBScan is used for the initial clustering of users’ dataset according to demographic information and deep neural network algorithm in order to classify new users. In this phase, all users are first clustered based on demographic information such as age, occupation and gender using the DBScan algorithm. For all users in the system, a
label is specified as a cluster. In phase 1, the content-based recommender system clusters users and assigns them to categories.

Initial Clustering of Users with DBScan Algorithm

At this point, the DBScan clustering algorithm separates users and determines the clusters based on demographic information of users. One of the most important features of the DBScan clustering algorithm compared to other clustering algorithms such as K-Means and X-Means, etc., is that the algorithm identifies and separates heterogeneous data. This reduces the complication of the model and in addition to improving the processing speed, improves the initial clustering process of users in the online movie system. DBScan clustering algorithm is a density-based spatial clustering algorithm that can also define anomalies in the dataset. DBScan clustering algorithm requires two user-defined parameters: Epsilon Proximity Distance (EPS) and the minimum number of minpts. For a given point, points in the eps distance are called adjacent points of that point. If the number of adjacent points is greater than minpts, this group of points is called cluster.

DBScan clustering algorithm labels data points as prime points, boundary points and remote (anomalous) points. eps has the lowest rating. The pseudo-code of DBScan clustering algorithm is given in the following algorithm. The inputs of this algorithm are the user-defined datasets and parameter values of eps and minpts. The following is a pseudo-code of the DBScan algorithm.
In algorithm (1), between lines 2 and 18, a for loop is executed for the number of points in the dataset. In step 2, DBSCAN assigns D [i] as the focal point. In step 3, the distance between the focal point D [i] and the remaining points is calculated, and then in step 4 points whose distance is less than or equal to EPS are accepted as adjacent points of the focal point. In step 5, the number of adjacent points of the focal point is calculated. Step 6 of the algorithm checks if the main point is on the adjacent points' list. In step 7, the algorithm checks the number of adjacent points in order to use the largest or equal minpt. If so, the center is designated as a main point. Between lines 8 and 12, a unique class is assigned to the focal point D [i] and the adjacent users. If the focal point is not a focal point but close to a main point, we define it as a focal point between steps 13 to 15. If the focal point is neither a main point nor a boundary point and the distance of points from eps is greater, we consider it as out-of-range data. Finally, in line 19, the algorithm presents the results. Table 2 shows the demographic information of 5 users.
Table 2. Demographic information of 5 users

| User code | Gender | Age | Job |
|-----------|--------|-----|-----|
| 1         | 1      | 56  | 16  |
| 2         | 1      | 25  | 15  |
| 3         | 1      | 45  | 7   |
| 4         | 1      | 25  | 20  |
| 5         | 2      | 50  | 9   |

As can be seen in Table (2), the second column shows the gender, the third column shows the age and the fourth column shows the users’ occupation according to the table below. Table 3 shows the values of the jobs defined in the users’ demographic file.

Table 3. The values of the jobs defined in the user demographic file

|   |   |
|---|---|
| 0: | "other" or not specified |
| 1: | "academic/educator" |
| 2: | "artist" |
| 3: | "clerical/admin" |
| 4: | "college/grad student" |
| 5: | "user service" |
| 6: | "doctor/health care" |
| 7: | "executive/managerial" |
| 8: | "farmer" |
| 9: | "homemaker" |
| 10: | "K-12 student" |
| 11: | "lawyer" |
| 12: | "programmer" |
| 13: | "retired" |
| 14: | "sales/marketing" |
| 15: | "scientist" |
| 16: | "self-employed" |
| 17: | "technician/engineer" |
| 18: | "tradesman/craftsman" |
| 19: | "unemployed" |
| 20: | "writer" |

As we see, each occupation has a code. The number of key users clustered in the current study is 6,040. Therefore, after applying DBScan clustering algorithm, the results can be presented in Table (4).
After applying the clustering algorithm, all users are clustered based on demographic information (age, gender, and occupation). Each user is assigned a cluster label. These labels are used as categories for each user. After clustering, users should be divided into two groups. The first category, which accounts for 70% of all users, is used to train deep neural networks and generate models. The second category includes new users who constitute 30% of all users.

**Separation of training and test samples**

One of the important phases to train the deep neural network algorithm is dividing samples into two main parts. The first part is used to train the deep neural network algorithm model and the second part is used as a test case for categorizing new users. Sampling is one of the stages of data mining which considered in the proposed solution.

**Classification of New Users with Deep Neural Network**

At this stage, using the demographic information of the new users and the clusters specified in the previous step, the appropriate category for the new users can be found. Fig. 5 shows the steps of cluster selection using deep neural network method.

| User code | Gender | Job | occupation | Cluster     |
|-----------|--------|-----|------------|-------------|
| 1         | 1      | 56  | 16         | cluster_1   |
| 2         | 1      | 25  | 15         | cluster_2   |
| 3         | 1      | 45  | 7          | cluster_3   |
| 4         | 1      | 25  | 20         | cluster_2   |
| 5         | 2      | 50  | 9          | cluster_1   |
As can be seen in Fig. 5, the training data that is the output of the clustering stage is processed by deep neural network method and the desired model is generated. Then the new user enters the system as the test data and is assigned to a category. When the cluster or category of the new is determined, its adjacent users which include the users of that category are extracted. The comments of the adjacent users are taken into consideration in movie recommendations. One of the most important reasons for using deep neural network algorithm is to support a large number of hidden layers and high classification accuracy when having huge amount of data.

**Phase 2: Collaborative Recommender System Based on Hybrid Similarity Criterion**

After determining the category for new users, phase 2 of the proposed method begins. The phase includes a collaborative recommender system based on new similarity criteria. In this phase, similarities are calculated based on a threshold (lambda) between the new user and the users in the selected category. Similarity criteria are determined based on age, gender, and
occupation. The collaborative recommender system extracts users who are the most similar to the new user. Then the higher rated movie services are suggested to the new user based on adjacency matrix.

Suppose that the total number of users in the system is shown as \( U = \{ u_1, u_2, u_3, ..., u_m \} \), new users are as \( N = \{ n_1, n_2, n_3, ..., n_n \} \), Collection of products as \( I = \{ i_1, i_2, i_3, ..., i_k \} \), and the demographic information of users as \( D = \{ d_1, d_2, d_3, ..., d_l \} \). Therefore, once the target cluster or category is selected for the new user, adjacent users who are similar to that user \( n_j \in N \) should be extracted. The corresponding algorithm for finding the nearest adjacent users is shown in Algorithm (2) in the next section.

```
1. public static List<Users> adjacency_Calculation(Users U)
2. { 
3.     List<Users> lstneighbors = new List<Users>();
4.     string Cnj = U.Cluster;//Cluster For New user
5.     foreach (Users item in Other.Users)
6.     { 
7.         string Cuj = item.Cluster;
8.         if (Cnj == Cuj)
9.             { 
10.             lstneighbors.Add(item);
11.         }
12.     }
13.     return lstneighbors;
14. }
```

Algorithm 2. The corresponding algorithm to find the nearest adjacent users

As can be seen in the first line, the number of users is entered in the algorithm as input. Line three defines the list the users. In line four, a category is defined for the new user. In lines 5 through 12, users near to the new user are calculated. Therefore, when adjacent users are initialized in NG format, the similarity between the new users and the users in the NG list should be calculated. Finally, the ultimate prediction is based on the rates of the adjacent users. The results of the similarity criteria are based on the demographic characteristics of the users as indicated by the set “D”. The final equation for calculating the similarity between
new users and adjacent users is obtained using the following equation.

\[
\text{sim}(n, u) = \frac{\sum_{j=1}^{n} SF_j \cdot w_j}{\sum_{j=1}^{n} w_j}
\]  \hspace{1cm} (1)

Where \(SF_j\) is the similarity value of the characteristic \(j\) and \(w_j\) is the weight of the attribute in question. In the present study, we have used characteristics like age, gender, and occupation to cluster users and calculate their similarities and based on their importance they are assigned with different weights. For example, 0.5 is assigned to age, 0.25 is assigned to gender and 0.25 is assigned to occupation. These values can also be varied but the total sum must be equal to 1. So:

\[D = \{d_1 = \text{age}, d_2 = \text{gender}, d_3 = \text{Occupation}\}\] And the set of weights are: \(W = \{w_1 = 0.5, w_2 = 0.25, w_3 = 0.25\} = 1\).

In this paper, a hybrid criterion is used to calculate the similarity between users. For each characteristic \(d_j\) we have defined an SF function \((a_{1}, a_{2})\) with values between \([0,1]\). This function calculates the similarity of two characteristics associated with a pair of users. Given the nature of the characteristics that we consider to calculate similarities, there are two general groups of features:

**Numeric features**

For numeric features such as age, a similarity criterion is defined as follows (2):

\[
w_{\text{age}} = (1 - \frac{|\text{Diff}|}{\text{Diff}_{\text{max}}})^\beta \quad \text{if } |\text{Diff}| \leq \text{Diff}_{\text{max}}
\]

\[
0 \quad \text{if } |\text{Diff}| > \text{Diff}_{\text{max}}
\]  \hspace{1cm} (2)

In the equation above, \(\text{Diff}\) represents the age difference between users and \(\text{Diff}_{\text{max}}\) is the maximum difference defined by the researcher. If the researcher wants to increase the value of \(w_{\text{age}}\) to the value of \(\text{Diff}\), he should simply set the value of \(\beta\) less than 1.

**String features**
In equation (3), the equation for calculating the similarity based on the string features is shown.

\[ W_{\text{gender}} = \begin{cases} 1 & \text{if } \text{att1} == \text{att2} \\ 0 & \text{if } \text{att1} <> \text{att2} \end{cases} \]  

(3)

Considering that whether the feature values of 1 and 2 are the same or not, the value of 1 or 0 is reset.

**Formation of Adjacency Matrix**

After obtaining the similarity level of the new user to other adjacent users using the similarity criteria, the adjacency matrix associated with the rates given by the adjacent users should be created for the ratings and through using a prediction formula which will be further explained, the ratings given by each user to the desired product should be calculated with the similarities obtained in the previous step and predict the rating as the success factor of the target product in order to recommend it to the new user. It should be noted that the highest rated products are recommended as superior products. The general form of the adjacency matrix for a product and user is as follows.

\[
\begin{array}{cccccccc}
I_1 & I_2 & I_3 & I_4 & I_5 & I_6 & I_7 & I_b \\
u1 & 5 & 4 & 4 & 5 & 2 & 5 & 5 & 3 \\
u2 & 3 & 5 & 5 & 4 & 5 & 4 & 5 & 5 \\
u3 & 4 & 3 & 5 & 4 & 3 & 3 & 4 & 4 \\
u4 & 5 & 5 & 5 & 5 & 2 & 2 & 4 & 3 \\
u5 & 1 & 2 & 4 & 3 & 1 & 3 & 4 & 5 \\
u6 & 2 & 5 & 3 & 2 & 3 & 4 & 5 & 4 \\
u7 & 5 & 4 & 5 & 1 & 5 & 5 & 5 & 5 \\
j & & & & & & & & \\
\end{array}
\]  

(4)

As stated in relation (4), user 1 gives item 1 a rating of 5 and rating 3 to item 2. In this article we simulate all rated items in order to generate the adjacency matrix of all users in the adjacent list and the new user and then we predict the desired value for the new user in the list.
Predicting new user’s rating

After forming the adjacency matrix in the previous step, the value of rates given by the users should be calculated for the new user and presented as prediction. So, we make predictions for the new user in the final phase. For each nj user, the proposed model must predict values for the item Ib. \( R_{nj,ib} \) is a predicted rating assigned to item b by the new user. The predicted rating for each user is obtained using the following equation.

\[
R_{nj,ib} = \frac{\sum_{u \in NG} \text{Sim}(nj,u) r_{u,ib}}{\sum_{u \in NG} \text{Sim}(nj,u)} + TF
\]  

(5)

\( r_{u,ib} \) is the rating given to item i by the user u in adjacent users’ list. Therefore, using the aforementioned prediction formula, the predicted ratings for each item will be approximately predicted the prediction which has the highest rating will be selected. The TF value that indicates the new or old user is effective in the provided rating. If the user is new, the value is \( TF = 0 \) and the doubled rating given by the researcher tends to vary from 0.1 to 1, which may affect the predicted total rating.

After defining the movie services for the new user, phase 3 which is FriendLink Algorithm, begins. At this point, the movies are transmitted through user communication dataset. In this phase, movies which are based on the relationship between those users who are similar to the new user are selected. So far, different similarity criteria have been proposed to calculate the similarity between users X and Y in graph G of social networks. Most of these criteria use the degree of the nodes and their adjacency in the network to calculate overall similarities based on the link. The proposed similarity criterion is based on the following four key factors in the social network graph. These factors include:

- Degree of the nodes
- User popularity
- Number of routes
Node Balance

Based on these factors, relation (6) is defined:

\[
Sim_{MyApp}(X,Y) = AVG\left(\frac{|Neib(Y)|}{N}\right) + \frac{|Paths_{X,Y}^{i}|}{|Paths_{X,Y}|} + \frac{K_Y}{\sum_{i=1}^{n} D(K_i)} + \frac{|Neib(X) \cap Neib(Y)|}{Avg(K_X, K_Y)}
\] (6)

In the above relationship, \(Sim_{MyApp}(X,Y)\) indicates the proposed similarity criterion for predicting links in social networks. \(|Neib(Y)|\) denotes the number of adjacent users of node Y. \(N\) is the sum of all nodes in the graph. \(|Paths_{X,Y}^{i}|\) indicates the number of paths with lengths \(L\) from node X to node Y in graph G. \(|Paths_{X,Y}|\) is the total number of paths leading to node Y. \(K_Y\) is the degrees of the node Y. \(\sum_{i=1}^{n} D(y_i)\) is the total graph degree, \(n\) is the number of nodes of graph G. \(|Neib(X) \cap Neib(Y)|\) is the number of shared adjacent users of node X and node Y. \(Avg(K_X, K_Y)\) is the average degree of node X and node Y. Finally, relation (6) can be used to obtain the degree of similarity between the users X and Y. One of the most important features of this criterion is that it calculates the similarity between two users on social network with great accuracy. This level of accuracy makes recommended users or friends more attractive.

**Phase 3: Improved Friendlink Algorithm**

In this phase, which is almost the ultimate phase, the improved Friendlink algorithm is run on the dataset to calculate the similarity between users who are connected though the link. In this section, we first describe the Link Prediction via Friendlink and the improved algorithm. Friendlink algorithm is a link prediction algorithm that is widely used to predict future links, especially dating, on social networks.

\[
Sim(n_i, u_j) = \sum_{i=2}^{l} \frac{1}{i-1} \cdot \frac{|path_{n_i,u_j}^i|}{\prod_{y=2}^{n-y}}
\] (7)

\(n_i\) is a new user who has recently entered the social network graph and \(u_j\) is the target user.

\(L\): Specifies the path length.
n: Specifies the total number of graph nodes.

\[ \sum_{i=2}^{L} \frac{1}{i-1} \] It is a weighting factor which is more effective for paths whose length is more than \( L = 2 \). Suppose that the maximum path length is 2 (\( L = 2 \)). In this case, \( \sum_{i=2}^{L} \frac{1}{i-1} = 1 \) and has no effect on the ultimate similarity. When \( L = 3 \), this weight is changed to 0.5 and has a significant effect. \( |\text{path}_{n,i}^{u,j}| \) is the number of paths between the origin and destination users which is shown by their length. Consider the following figure. Assuming that the new user is \( u_1 \) and the associated users are \( u_2, u_3, u_5, u_6, u_8 \), the maximum path length to \( u_4 \) that is a friend user suggested to the new user (\( u_2, u_3 \)) is 3. \( \prod_{y=2}^{L} (n - y) \) is the number of possible paths from new user to target user. One of the most important advantages of using this similarity criterion is that it does not stop at lengths 1 and continues running to length 2, 3, and 4 in the graph. Consider Figure (6).

As shown in the figure, the origin user (green arrow) which is located in the corresponding graph has 5 (2, 3, 5, 6 and 8) direct friends. The basic algorithm of Friend Link first obtains the graph of the whole network and after calculating the adjacent users of the source node, calculate the best paths to reach the other nodes considering the path length and finally according to the presented formula, calculates the similarity. We have also improved Friendlink formula to improve the accuracy of link prediction. Applying the adjacency degree of the nodes the formula will change as follows.
\[
Sim(n_j, u_j) = \left( \sum_{i=2}^{l} \frac{1}{\prod_{y=2}^{l-1}(n-y)^i} \right) \times \left( \frac{D_{u_j}}{N} \right)
\] (8)

\(N\) is the total number of degrees of nodes. \(D_{u_j}\) is the degree of the target node. So, we try to improve the results of the article by using the Friendlink algorithm and improving the similarity formula. In the proposed method, the presented method is called Pro-Friendlink. The important thing about Pro-Friendlink algorithm is that it gives more importance to the friends that are more connected to the users and ultimately recommend users who have more credibility in the social network to the target user. It makes it possible to recommend more attractive and popular users to the new users.

An example of Pro-Friendlink algorithm is presented here. Pro-Friendlink prediction method calculates the similarity between nodes in a unidirectional graph so that users’ credibility is taken into account. Pro-Friendlink algorithm receives G-graph communications as input, and after generating the adjacency matrix, calculates the similarity between the two nodes and indicates it as output. Consequently, the friend suggestions can be based on the weights calculated by Pro-Friendlink prediction algorithm in the adjacency matrix. Figure 3 shows the Pro-Friendlink prediction algorithm. If we want to suggest a new friend to user U1, there is no direct indication of this in the adjacency matrix shown in Table (1). After running Pro-Friendlink algorithm, we can find the similarity matrix between the two nodes of the G graph and suggest friends based on importance.

In the proposed method, we first modify the adjacency matrix \(A\) displayed in Table (2). So that instead of having values of 0 and 1, the input (i,j) is a list of paths from node i to node j. The basic idea is that if the adjacency matrix \(A\), which contains 0/1 of a graph, is increased to a power \(N\) by the adjacency matrix, then the result of the input data (i, j) shows how long the path from node VI to node VJ There is. Then, instead of counting the routes, we will look for all real routes.
**Phase 4: Combining Link System and Recommender System**

Phase 4 of the proposed method is related to the combination of collaborative recommender system’s output and improved Friendlink algorithm. In Phase 4, which is the final phase in the proposed system, the results of Phases 1 and 2 are combined with Phase 3, and the jointly selected users are sent as the final proposal. Suppose that the results of Phase 1 and 2 were users 1, 2, 4, 5, 6, 9 and 10. the results of phase 3 were users 3, 4, 5, 6, 9 and 10. The results are both combined and users 4, 5, 6, 9, 10 are suggested as attractive users to the new user.

**Results**

In this paper, simulation on Movielens dataset is performed to investigate the issue and evaluate the results. To access the data source used, simply refer to [22] and select and download the desired data from the versions provided. The version used in this study was 2013 whose size is 1 MB. These files contain 1,000,209 recordings of user ratings, 3,900 movie samples and 6,040 user samples, with each user rating at least 20 movies. Mean Absolute Error and Root Mean Square are used as evaluative criteria.

\[
MAE = \frac{1}{k} \sum |P_{u,i} - r_{u,i}|
\]

\[
RMSE = \sqrt{\frac{1}{k} \sum (P_{u,i} - r_{u,i})^2}
\]

Where \( P_{u,i} \) is the predicted rating of user \( u \) to the movie \( i \) and \( r_{u,i} \) is the actual rating of user \( u \) to the movie \( i \). Here are some of the scenarios that are illustrated in the table below.

| Weight       | Scenarios |
|--------------|-----------|
| W1=0.6, W2=0.3, W3=0.1 | Scenario 1 |
| W1=0.3, W2=0.6, W3=0.1 | Scenario 2 |
| W1=0.3, W2=0.1, W3=0.6 | Scenario 3 |
As can be seen, different results are obtained according to different weightings for age, sex, and occupation. The above scenarios are defined with different weights. Weights that have more features are actually more focused on the feature and have more similarity effects. The following table calculates the mean prediction errors of the proposed method using MAE and RMSE and compares them with that of other methods. The stage considered for the following results is stage 1.

Table 6. Comparison of MAE, Root RMSE of Proposed Method with Other Methods = 100 (Step 1)

| Methods                  | MAE  | RMSE |
|--------------------------|------|------|
| Decision tree            | 0.86 | 1.09 |
| Multifaceted decision tree| 0.9  | 1.12 |
| Naïvebays                | 0.89 | 1.13 |
| Random Classification    | 0.92 | 1.19 |
| proposed method          | 0.35 | 0.59 |

As can be seen, the proposed method with 100 users as input has an MAE of 0.35 and a RMSE of 0.59. Therefore, the proposed method is much more accurate than the other methods outlined in the table above. The following table calculates the mean prediction errors using MAE and RMSE in the proposed method and compares it to the other methods considering Step 3.

Table 7. Comparison of MAE and Root RMSE of Proposed Method with Other Methods = 100 (Step 2)

| Methods                  | MAE  | RMSE |
|--------------------------|------|------|
| Decision tree            | 0.86 | 1.09 |
| Multifaceted decision tree| 0.91 | 1.15 |
| Naïvebays                | 0.89 | 1.14 |
| Random Classification    | 0.92 | 1.2  |
| proposed method          | 0.35 | 0.59 |

As can be seen, the proposed method with 100 users as input has an MAE of 0.35 and a RMSE of 0.59. Therefore, the proposed method is much more accurate than the other methods outlined in the table above. The following table calculates the mean prediction errors
using MAE and RMSE in the proposed method and compares it to the other methods considering Step 1.

Table 8. Comparison of MAE and Root RMSE of Proposed Method with Other Methods with Number of Users=100 (Step 3)

| Methods                   | MAE  | RMSE |
|---------------------------|------|------|
| Decision tree             | 0.86 | 1.09 |
| Multifaceted decision tree| 0.9  | 1.14 |
| NaïveBayes                | 0.9  | 1.14 |
| Random Classification     | 0.92 | 1.29 |
| proposed method           | 0.35 | 0.59 |

As can be seen, the proposed method with 100 users as input has an MAE of 0.35 and a mean square error of 0.59. Therefore, the proposed method is much more accurate than the other methods outlined in the table above. The following table calculates the mean prediction errors using MAE and RMSE in the proposed method and compares it to the other methods considering Step 1.

Table 9. Comparison of MAE and Root RMSE in Proposed Method with Other Methods = 500 (Step 1)

| Methods                | MAE  | RMSE |
|------------------------|------|------|
| Decision tree          | 0.83 | 1.03 |
| Multifaceted decision tree | 0.845 | 1.09 |
| NaïveBayes             | 0.835| 1.03 |
| Random Classification  | 0.87 | 1.1  |
| proposed method        | 0.76 | 1.03 |

As can be seen, the proposed method with 500 users as input has an MAE of 0.76 and a RMSE of 1.03. Therefore, the proposed method is much more accurate than the other methods outlined in the table above. The following table calculates the mean prediction errors using MAE and RMSE in the proposed method and compares it to the other methods considering Step 2.
Table 10. Comparison of MAE and Root RMSE of Proposed Method with Other Methods = 500 (Step 2)

| Methods                  | MAE  | RMSE |
|--------------------------|------|------|
| Decision tree            | 0.83 | 1.06 |
| Multifaceted decision tree | 0.85 | 1.09 |
| Naïvebays                | 0.839| 1.06 |
| Random Classification    | 0.87 | 1.1  |
| proposed method          | 0.76 | 1.03 |

As can be seen, the proposed method with 500 users as input has an MAE of 0.76 and a RMSE of 1.03. Therefore, the proposed method is much more accurate than the other methods outlined in the table above. The following table calculates the mean prediction errors using MAE and RMSE in the proposed method and compares it to the other methods considering Step 3.

Table 11. Comparison of MAE and Root RMSE of Proposed Method with Other Methods with Number of Users = 500 (Step 3)

| Methods                  | MAE  | RMSE |
|--------------------------|------|------|
| Decision tree            | 0.83 | 1.06 |
| Multifaceted decision tree | 0.86 | 1.08 |
| Naïvebays                | 0.84 | 1.06 |
| Random Classification    | 0.87 | 1.09 |
| proposed method          | 0.76 | 1.03 |

As can be seen, the proposed method with 500 users as input has an MAE of 0.76 and a RMSE of 1.03. Therefore, the proposed method is much more accurate than the other methods outlined in the table above. The following table calculates the mean prediction errors using MAE and RMSE in the proposed method and compares it to the other methods considering Step 1.

Table 12. Comparison of MAE and Root RMSE of Proposed Method with Other Methods with Number of Users = 900 (Step 1)

| Methods      | MAE  | RMSE |
|--------------|------|------|
| Decision tree | 0.83 | 1.01 |
As can be seen, the proposed method with 900 users as input has an MAE of 0.73 and a RMSE of 0.95. Therefore, the proposed method is much more accurate than the other methods outlined in the table above. The following table calculates the mean prediction errors using MAE and RMSE in the proposed method and compares it to the other methods considering Step 2.

Table 13. Comparison of MAE and Root RMSE of Proposed Method with Other Methods = 900 (Step 2)

| Methods                     | MAE  | RMSE  |
|-----------------------------|------|-------|
| Decision tree               | 0.82 | 1.01  |
| Multifaceted decision tree  | 0.82 | 1.01  |
| Naïvebays                   | 0.82 | 1.01  |
| Random Classification       | 0.82 | 1.04  |
| proposed method             | 0.73 | 0.95  |

As can be seen, the proposed method with 900 users as input has an MAE of 0.73 and a RMSE of 0.95. Therefore, the proposed method is much more accurate than the other methods outlined in the table above. The following table calculates the mean prediction errors using MAE and RMSE in the proposed method and compares it to the other methods considering Step 2.

Table 14. Comparison of MAE and Root RMSE in Proposed Method with Other Methods = 900 (Step 3)

| Methods                     | MAE  | RMSE  |
|-----------------------------|------|-------|
| Decision tree               | 0.82 | 1.01  |
| Multifaceted decision tree  | 0.82 | 1.01  |
| Naïvebays                   | 0.82 | 1.01  |
| Random Classification       | 0.82 | 1.04  |
| proposed method             | 0.73 | 0.95  |
As can be seen, the proposed method with 900 users as input has an MAE of 0.73 and a RMSE of 0.95. Therefore, the proposed method is much more accurate than the other methods outlined in the table above.

Finally, the table below summarizes the evaluation of the proposed method with 100 users and comparing it with other methods using MAE.

Table 15. Evaluation of the proposed method with other methods with 100 users

|                | 100 users - Step 1 | 100 users - Step 2 | 100 users - Step 3 |
|----------------|---------------------|---------------------|---------------------|
| decision tree  | MAE: 0.86, RMSE: 1.09 | MAE: 0.83, RMSE: 1.03 | MAE: 0.86, RMSE: 1.09 |
| Multifaceted decision tree | MAE: 0.9, RMSE: 1.12 | MAE: 0.845, RMSE: 1.09 | MAE: 0.9, RMSE: 1.14 |
| Naivebayes     | MAE: 0.89, RMSE: 1.13 | MAE: 0.835, RMSE: 1.03 | MAE: 0.9, RMSE: 1.14 |
| Random Classification | MAE: 0.92, RMSE: 1.19 | MAE: 0.87, RMSE: 1.1 | MAE: 0.92, RMSE: 1.29 |
| proposed method| MAE: 0.35, RMSE: 0.59 | MAE: 0.76, RMSE: 1.03 | MAE: 0.35, RMSE: 0.59 |

The following table also summarizes the evaluations of steps 1, 2 and 3 with 500 users.

Table 16. Evaluations of steps 1, 2 and 3 with 500 users

|                | 500 users - Step 1 | 500 users - Step 2 | 500 users - Step 3 |
|----------------|---------------------|---------------------|---------------------|
| decision tree  | MAE: 0.83, RMSE: 1.03 | MAE: 0.83, RMSE: 1.06 | MAE: 0.83, RMSE: 1.06 |
| Multifaceted decision tree | MAE: 0.845, RMSE: 1.09 | MAE: 0.85, RMSE: 1.09 | MAE: 0.86, RMSE: 1.08 |
| Naivebayes     | MAE: 0.835, RMSE: 1.03 | MAE: 0.839, RMSE: 1.06 | MAE: 0.84, RMSE: 1.06 |
| Random Classification | MAE: 0.87, RMSE: 1.1 | MAE: 0.87, RMSE: 1.1 | MAE: 0.87, RMSE: 1.09 |
| proposed method| MAE: 0.76, RMSE: 1.03 | MAE: 0.76, RMSE: 1.03 | MAE: 0.76, RMSE: 1.03 |

The following table also summarizes the evaluations of steps 1, 2 and 3 with 900 users.

Table 17. Evaluations of steps 1, 2 and 3 with 900 users

|                | 900 users - Step 1 | 900 users - Step 2 | 900 users - Step 3 |
|----------------|---------------------|---------------------|---------------------|
| decision tree  | MAE: 0.83, RMSE: 1.01 | MAE: 0.82, RMSE: 1.01 | MAE: 0.82, RMSE: 1.01 |
Based on the evaluation of the proposed method and comparing it with other methods of research conducted in 2014 and 2015, the diagram below compares MAE of the proposed method with other basic methods such as decision tree algorithms, Naïvebays, Random Classification [20] as well as algorithms like SVD and ApproSVD [26].

In other reviews we enable our paper with [24]. Chu-Hsing Lin et. al, in [24], presented the neural network for movie recommendation system. Table. 6, shows the comparison of the proposed method using boosting approach with Scikit-learn, TensorFlow. Scikit-learn is an early machine learning tool with many ready-made libraries and functions to call, and has powerful and fast mathematics capabilities. TensorFlow is a tool that has emerged in recent years with the development of deep learning.

Table 18. comparison of the proposed method using boosting approach with Scikit-learn, TensorFlow

| Method         | Type            | MAE  | Process Time (s) |
|----------------|-----------------|------|------------------|
| Scikit-learn   | Normal RS       | 0.88 | 31.49            |
| TensorFlow     |                 | 0.88 | 104.336          |
| Scikit-learn   | Neural network  | 1.43 | 6264.01          |
| TensorFlow     | RS              | 0.76 | 136.28           |
| MyApproach     | Boosting        | 0.35 | 98.021           |

As can be seen in Table (18), the proposed method performs better than Scikit-learn and TensorFlow methods with and without applying neural network. Also, this method has a lower processing time comparing with other methods.

Discussion
The main purpose of this article is to solve cold start problem in online movie networks and to introduce appropriate movies to new users with acceptable accuracy. To do this, content-based recommender systems and collaborative filtering as well as clustering techniques and deep neural network were used. In this article, the researcher applied clustering techniques, deep neural network, hybrid similarity criteria, and improved Friend link algorithm as methods which are much more accurate than any other methods used before to provide new users with appropriate movies. Therefore, given the simulation to provide attractive movies to new users who are experiencing a cold start, the proposed method, compared to other methods, recommends desired movies to users in a timely manner. So another major issue is the trust issue that arises from disregarding older users. The proposed method consists of four steps:

1) initial clustering of all users and assigning new users to appropriate clusters;

2) assigning appropriate weights to the characteristics of the target cluster users and determining the adjacent users

3) forming adjacency matrix of adjacent users' ratings to the existing movies and calculating the new user's ratings considering adjacent users' ratings and similarity level that exists between the users. At this stage doubled rating opportunity is created for loyal users.

4) Using Friendlink algorithm to introduce similar users to the new user and to combine Step 4 with Step 3.

Finally, to evaluate the prediction error of the proposed method in comparison with other similar methods such as C24.5, CM4.5, RCA and Naïvebayes method, MAE and RMSE evaluation criteria were used. The error rate of the proposed method is less than other similar methods and has indicated acceptable accuracy in introducing movies to new users.

Conclusion
In our future work, we would like to focus on several areas. Here are some recommendations for further research: 1) in the present article, the k-mean clustering technique is used to cluster users with the number of k-tests obtained by trial and error. Therefore, techniques such as random-walk clustering algorithm or improved clustering algorithm can be used and the results can be compared and evaluated with the current and other methods of clustering. 2) In this paper, the idea of deep neural network is used to assign new users to the desired cluster to select the categories that are most suitable for users. Therefore, future works can replace other techniques and compare the results with the proposed method. This method can be implemented for optimal operation by fog computing in distributed models.

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Availability of data and materials

The datasets generated and/or analysed during the current study are not publicly available due [REASON WHY DATA ARE NOT PUBLIC] but are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests

Funding

Not applicable

Authors’ contributions
All authors contributed to developing the ideas, and writing and reviewing this manuscript. All authors read and approved the final manuscript.

Acknowledgement

Not applicable

References

1. Eirinaki M, Gao J, Varlamis I, Tserpes K. Recommender systems for large-scale social networks: A review of challenges and solutions.
2. Najafabadi MK, Mohamed AH, Mahrin MN. A survey on data mining techniques in recommender systems. Soft Computing. 2019 Jan 30;23(2):627-54.
3. Silveira T, Zhang M, Lin X, Liu Y, Ma S. How good your recommender system is? A survey on evaluations in recommendation. International Journal of Machine Learning and Cybernetics. 2019 May 1;10(5):813-31.
4. Tatiana K, Mikhail M. Market basket analysis of heterogeneous data sources for recommendation system improvement. Procedia Computer Science. 2018 Jan 1;136:246-54.
5. Hoang L. HU-FCF++: A novel hybrid method for the new user cold-start problem in recommender systems. Eng. Appl. Artif. Intell.. 2015;41:207-22.
6. Pera MS, Ng YK. A group recommender for movies based on content similarity and popularity. Information Processing & Management. 2013 May 1;49(3):673-87.
7. Christensen IA, Schiaffino S. Entertainment recommender systems for group of users. Expert Systems with Applications. 2011 Oct 1;38(11):14127-35.
8. Camacho LA, Alves-Souza SN. Social network data to alleviate cold-start in recommender system: A systematic review. Information Processing & Management. 2018 Jul 1;54(4):529-44.
9. Van Meteren R, Van Someren M. Using content-based filtering for recommendation. In Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop 2000 May 30 (Vol. 30, pp. 47-56).

10. Basu C, Hirsh H, Cohen W. Recommendation as classification: Using social and content-based information in recommendation. In Aaai/iaai 1998 Jul 1 (pp. 714-720).

11. Chen J, Wang H, Yan Z. Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering. Swarm and Evolutionary Computation. 2018 Feb 1;38:35-41.

12. Benkhelifa R, Bouhyaoui N, Laallam FZ. A demographic-based approach for improved content categorization in social networking. In 2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP) 2018 Apr 25 (pp. 1-5). IEEE.

13. Nirenburg S, Carbonell J, Tomita M, Goodman K. Machine translation: A knowledge-based approach. Morgan Kaufmann Publishers Inc.; 1994 Jan 1.

14. Ha T, Lee S. Item-network-based collaborative filtering: A personalized recommendation method based on a user's item network. Information Processing & Management. 2017 Sep 1;53(5):1171-84.

15. Sharma L, Gera A. A survey of recommendation system: Research challenges. International Journal of Engineering Trends and Technology. 2013;4(5):1989–92. http://www.ijettjournal.org/archive/ijett-v4i5p132

16. Ghazanfar MA, Prugel-Bennett A. A scalable, accurate hybrid recommender system. In 2010 Third International Conference on Knowledge Discovery and Data Mining 2010 Jan 9 (pp. 94-98). IEEE.

17. Kim HN, El-Saddik A, Jo GS. Collaborative error-reflected models for cold-start recommender systems. Decision Support Systems. 2011 Jun 1;51(3):519-31.
18. Bobadilla J, Ortega F, Hernando A, Bernal J. A collaborative filtering approach to mitigate the new user cold start problem. Knowledge-based systems. 2012 Feb 1;26:225-38.

19. Byström H. Movie Recommendations from User Ratings.

20. Lika B, Kolomvatsos K, Hadjiefthymiades S. Facing the cold start problem in recommender systems. Expert Systems with Applications. 2014 Mar 1;41(4):2065-73.

21. Pereira AL, Hruschka ER. Simultaneous co-clustering and learning to address the cold start problem in recommender systems. Knowledge-Based Systems. 2015 Jul 1;82:11-9.

22. MovieLens, GroupLens. 2015. http://grouplens.org/datasets/movielens/2013.

23. Kutty S, Chen L, Nayak R. A people-to-people recommendation system using tensor space models. InProceedings of the 27th Annual ACM Symposium on Applied Computing 2012 Mar 26 (pp. 187-192).

24. Lin CH, Chi H. A novel movie recommendation system based on collaborative filtering and neural networks. InInternational Conference on Advanced Information Networking and Applications 2019 Mar 27 (pp. 895-903). Springer, Cham.

25. Walek B, Fojtik V. A hybrid recommender system for recommending relevant movies using an expert system. Expert Systems with Applications. 2020 May 13:113452.

26. Zhou X, He J, Huang G, Zhang Y. SVD-based incremental approaches for recommender systems. Journal of Computer and System Sciences. 2015 Jun 1;81(4):717-33.