Pamukkale Univ Muh Bilim Derg, 30(2), 212-221, 2024

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

A hybrid multi-criteria recommendation algorithm based on autoencoders

Otokodlayıcılara dayalı hibrit çoklu-ölçütlü öneri algoritması

Zeynep BATMAZ[1*], *Cihan KALELİ*[1]

[1]Department of Computer Engineering, Faculty of Engineering, Eskisehir Technical University, Eskisehir, Turkey.
zozdemir@eskisehir.edu.tr, ckaleli@eskisehir.edu.tr

Received/Geliş Tarihi: 17.01.2023 Revision/Düzeltme Tarihi: 02.04.2023 doi: 10.5505/pajes.2023.68253
Accepted/Kabul Tarihi: 27.04.2023 Research Article/Araştırma Makalesi

Abstract

Multi-criteria recommender systems provide efficient solutions to deal with information overload problem by producing personalized recommendations considering multiple criteria. Even though multi-criteria recommender systems provide more accurate and personalized recommendations to their users compared with traditional recommender systems, sparsity becomes a major problem for such systems due to the increasing number of criteria. Due to the lack of co-rated items among users, finding out neighbors and producing accurate predictions become harder. Especially similarity-based multi-criteria recommendation approaches are significantly affected by the sparsity problem. Thus, aiming to minimize the negative impacts of that problem, a hybrid similarity-based multi-criteria recommendation method, that utilizes complex, low-dimensional and latent features obtained from both reviews and criteria ratings by autoencoders, is proposed in this work. The empirical results performed on a real data set with a sparsity percentage of 99.7235% show that the proposed work can provide more accurate predictions compared with other neighborhood-based multi-criteria approaches.

Keywords: Multi-criteria, Collaborative filtering, Autoencoders, Sparsity, Neighbor selection.

Öz

Çoklu-ölçütlü öneri sistemleri, aşırı bilgi sorunuyla başa çıkmak için birden fazla ölçütü dikkate alarak kişiselleştirilmiş öneriler üreterek etkili çözümler sunar. Çoklu-ölçütlü öneri sistemleri, geleneksel öneri sistemlerine göre kullanıcılarına daha doğru ve kişiselleştirilmiş öneriler sunsa da, artan kriter sayısı nedeniyle seyreklik bu tür sistemler için önemli bir sorun haline gelmektedir. Kullanıcılar arasında ortak puanlanan ögelerin olmamasından dolayı, komşuları bulmak ve doğru tahminler üretmek zorlaşmaktadır. Özellikle benzerlik-tabanlı çoklu-ölçütlü öneri yaklaşımları, seyreklik probleminden önemli ölçüde etkilenmektedir. Bu nedenle, bu çalışmada, bu sorunun olumsuz etkilerini en aza indirmek amacıyla, hem yorum hem de ölçüt derecelendirmelerinden otokodlayıcılar ile çıkarılan karmaşık, düşük boyutlu ve gizli özellikleri kullanan hibrit benzerlik-tabanlı çoklu-ölçütlü bir öneri algoritması önerilmiştir. Seyreklik yüzdesi %99,7235 olan gerçek bir veri seti üzerinde gerçekleştirilen deneysel sonuçlar, önerilen çalışmanın diğer komşuluk-tabanlı çok kriterli yaklaşımlara kıyasla daha doğru tahminler sağlayabildiğini göstermektedir.

Anahtar kelimeler:, Çoklu-Ölçüt, İşbirlikçi filtreleme, Otokodlayıcılar, Seyreklik, Komşu seçimi.

1 Introduction

Recommender systems are content filtering techniques which provide an impressive way of handling information overload problem. The service/product providers who utilize these systems take advantages of customer satisfaction, system maintainability and usability. Collaborative filtering (CF) is a recommendation technique which is based upon the thought that people who behave similarly will tend to be of the same mind evermore [1]. Over time, users have felt the need to evaluate services/products by considering several criteria as well as making general evaluations. This allows the customers to reach more personalized recommendations. Aiming to meet this need, multi-criteria collaborative filtering (MCCF) techniques allow the users to make evaluations by considering several criteria besides general evaluations [2]. Aggregation function- and similarity-based approaches are two basic groups of MCCF techniques [2]. Similarity-based approaches utilize the relationships among neighbors considering multiple criteria during the prediction process. Thus, the chosen neighbors and the similarities/distances among them directly affect the accuracy of the produced predictions.

First of all, criterion-based predictions are produced in the aggregation function-based methods. To compute the overall rating-based predictions, an aggregation function which shows the relationships between criteria and overall ratings is used.

Researchers show that utilizing ratings of multiple criteria in addition to a general evaluation during the prediction process improves accuracy of the produced recommendations [2],[3]. On the other hand, increasing number of criteria causes sparsity to be a major problem for MCCF systems [4]. Accuracy and the ratio of the produced predictions decrease with increasing sparsity level. Neighbor selection directly affects the prediction process in similarity-based MCCF approaches. The ratings belonging to the neighbors for an active item are determiner during the prediction process for that item. Because of the deficiency of the co-rated items among users, the neighbor selection that provides higher accuracy becomes harder. Thus, particularly similarity-based MCCF approaches are significantly affected by the sparsity problem.

Deep learning (DL) techniques have been used in many areas such as image processing [5], text mining [6], and recommender systems [7]. DL techniques have also recently been utilized frequently in recommender systems aiming to improve accuracy and deal with sparsity. Utilizing DL techniques in reducing dimensions of sparse user-item matrix [8] and extracting features from side information such as reviews or content information [9],[10] is a basic approach, especially for handling sparsity in traditional CF. Even though

*Corresponding author/Yazışılan Yazar

there are lots of approaches focusing on sparsity problem in MCCF systems, most of them are linear approaches that miss complex and unexpected features. Thus, producing recommendations over sparse user-item multi-dimensional matrix by eliminating the negative impacts of sparsity is still a challenge for MCCF systems.

Therefore, we improved the work [11] which provides pioneer accuracy results among similarity-based MCCF approaches by integrating extracted features from reviews into the recommendation process in this study. The proposed work AE-simMCCF++ is a hybrid similarity-based MCCF algorithm utilizing autoencoders in extracting features from both reviews of the users and multi-criteria ratings. With this way, AE-simMCCF++ improves the accuracy and the ratio of the number of produced predictions in similarity-based MCCF systems.

Contributions of the proposed method to the literature can be listed as pursues:

- A novel hybrid neighborhood-based MCCF algorithm depending on autoencoders is proposed,
- Complex, non-linear and unexpected features are extracted from reviews of users by autoencoders,
- Similarities among users/items are calculated using the hybridized low-dimensional complex, latent features obtained by both reviews and multi-criteria ratings.

The study is organized as follows: Existing solutions for dealing with sparsity issue in MCCF systems are given in Section 2. The fundamentals of a basic MCCF system and brief information about the used DL method are given in Section 3. Section 4 introduces the proposed method. The experimental works are presented in Section 5. The last section presents conclusions and future works.

2 Related work

With the rising number of criteria in multi-criteria recommender systems, sparsity has become a more prominent problem. The number of works that focus directly on solving the sparsity problem in MCCF systems is limited. Most of the available researches have transformed high-dimensional and sparse data into lower-dimensional and dense ones by reducing dimension in order to solve the sparsity problem in MCCF systems [12]-[14]. Moreover, in order to deal with the sparsity issue, hybrid MCCF techniques using taxonomy and ontological features extracted from the content data of the products in addition to the multi-criteria ratings have been proposed [15],[16]. In another approach, the preference-based similarity is used to find relationships among neighbors to solve sparsity issue [17]. In the work [18], social relationships among users are integrated into a multi-criteria recommender engine. All of these works are depending on linear approaches and ignore the knowledge which can be extracted from review information.

In traditional CF techniques, using the content or review information besides ratings is one of the most frequently used methods to deal with the sparsity problem [19]. In particular, the success of deep learning techniques in extracting hidden and complex features from data such as text, images, and signals with nonlinear methods and integrating these features into the systems have made these techniques frequently preferred in traditional recommender systems [20]-[24]. Especially extracting features from reviews with deep learning techniques and designing a hybrid recommender system is so popular in traditional CF systems. Even though the review data is smaller than the rating data in the real world, traditional recommender system approaches utilize reviews as extra information in hybrid systems. As a result, the approaches which utilize the reviews as a part of traditional hybrid recommender systems, improve the accuracy in sparse systems [25]-[27]. Even though deep learning techniques provide pioneer results in traditional CF systems with the ability of feature extraction and highly accurate classification, these abilities of deep learning techniques have not been used much in MCCF systems.

Studies that benefit from deep learning techniques in MCCF systems are limited and generally focus on accuracy problem. Autoencoders are used as an aggregation function in aggregation function-based MCCF systems to solve the accuracy issue [28]. AE-MCCF is another aggregation function-based MCCF algorithm which utilizes autoencoders and feed-forward neural networks to increase the accuracy of the produced predictions [29]. Another aggregation function-based MCCF algorithm focusing on solving the accuracy issue in MCCF systems uses multi-layer neural networks in both producing criterion-based ratings and learning the aggregation function [30]. All of these studies produce predictions using only features derived from raw user-item preference data. They ignore the possible features which can be obtained from content and/or review information. The features extracted from content information with stacked denoising autoencoders are integrated into tensor factorization [31]. Even though denoising autoencoders are used aiming to extract features from content information in the work [31], linear approaches are utilized in the prediction step. In the work [32], implicit ratings obtained from reviews with GloVe and Word2Vec are combined with explicit ratings to compute overall criterion ratings in aggregation function-based MCCF. In the work [33], user and item representations are learnt from auxiliary information aiming to incorporate them into tucker decomposition.

The existing studies generally focus on one dimension as accuracy and ignore other dimensions such as coverage. The solutions for improving accuracy are generally based on non-linear assumptions. Additionally, none of these studies provide any solution to the sparsity problem in similarity-based MCCF systems. Recently, a similarity-based MCCF algorithm utilizing autoencoders has been proposed aiming to deal with the sparsity problem [11]. Even though that work provides a solution for reducing the negative impacts of the sparsity issue in similarity-based MCCF systems, it only uses the extracted features from multi-criteria ratings. The effects of hidden knowledge, which can be obtained from content and/or review information, on the sparsity problem are ignored.

In this work, we improved the study [11] by integrating complex and hidden features obtained from users' reviews with autoencoders for dealing with the sparsity issue. Unlike other studies, aiming to reduce negative impacts of the sparsity issue, this study utilizes complex and non-linear features extracted by the autoencoders from both user reviews and multi-criteria ratings when selecting neighbors for similarity-based MCCF.

3 Background

3.1 Multi-criteria collaborative filtering

R: $Users\ x\ Items -> R_0\ x\ R_1\ x\ R_2\ x\ \ldots\ R_k$ represents users' preferences on items in MCCF systems. R_0 is the set of overall

ratings that users provide for the items. R_c is the set of ratings of the items provided by users in terms of the c^{th}criterion, with $c \in 1, 2, \ldots, k$.

Similarity-based MCCF systems use aggregated correlations among users/items in terms of overall and criteria ratings with the traditional neighborhood-based CF algorithm. Any similarity or distance measure can be utilized for calculating separate correlations between users'/items' preferences. Either average or worst-case similarity approaches are used to compute the aggregated similarities. In the worst-case similarity method, the minimum one of the calculated separate similarities is used as an aggregated similarity. Eq. (1) and Eq. (2) show the equations for the worst-case and average similarity computation between two users/items as u and t, orderly:

$$sim_{avg}(u,t) = \frac{\sum_{c=0}^{c=k} sim_c(u,t)}{k+1} \quad (1)$$

$$sim_{min}(u,t) = min_{c=0,\ldots,k}(sim_c(u,t)) \quad (2)$$

Where sim_c is considered as any similarity measure to compute the discrete correlations among users concerning the c^{th} criterion.

Aggregation function-based methods utilize an aggregation function which hangs off the relations between overall ratings and criteria ratings. Overall predictions are computed using the learned aggregation function and the discrete criterion-based predictions predicted by any traditional CF algorithm. The general equation for the prediction process is given in Eq. (3):

$$R_o = f\ (R_1, R_2, \ldots, R_k) \quad (3)$$

Wwhere R_o is the set of overall predictions and f is the aggregation function. $R_1, R_2, \ldots, R_k$ are the criterion-based predictions.

3.2 Autoencoders

An autoencoder is a feed-forward neural network whose aim is to reconstruct its input at the output layer [34]. An input layer, a fully connected hidden layer and an output layer are the basic layers which construct an autoencoder as given in Figure 1.

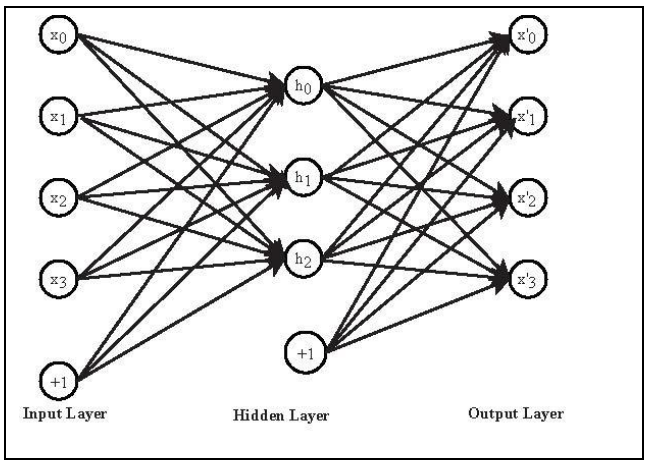

Figure 1. A simple autoencoder [29].

The input given in the input layer is encoded at the hidden layer as represented in Eq. (4). It is decoded at the output layer to obtain the original input data as shown in Eq. (5). The output of the encoding process can be benefited for reducing dimensions of high-dimensional data and extracting features. An autoencoder learns and trains its hyper-parameters by minimizing the loss function given in Eq. (6).

$$\mathrm{h(x)} = \theta(W_1 x + b_1) \quad (4)$$

Where θ, W_1, and b_1 are a non-linear function for the hidden layer, the weights between the input and hidden layer neurons and bias values for the hidden layer neurons, respectively., x represents the input of the autoencoder.

$$\mathrm{t(h(x))} = \delta(W_2 h(x) + b_2) \quad (5)$$

Where δ, W_2, and b_2 are a non-linear function for the output layer, the weights between the hidden and output layer neurons and bias values for the output layer neurons, respectively. $h(x)$ represents the encoded data.

$$\sum_{x \in X} \|x - t(h(x))\|_2^2 \quad (6)$$

4 The proposed approach

The proposed approach is a similarity-based hybrid multi-criteria recommendation approach that uses autoencoders to deal with the sparsity problem. The proposed approach includes two main stages as extraction of features and prediction steps. In the feature extraction stage, in addition to the low-dimensional, complex and dense features extracted from the users'/items' profiles by autoencoders with the AE-simMCCF approach [11], hidden and complex features are extracted by autoencoders from the reviews of the users about the items. With this purpose, an autoencoder is created for each criterion in order to extract features from rating-based users'/items' profiles. In order to extract features from users' reviews, firstly, reviews-based profiles of users/items are created. An autoencoder is created for the collection of review-based users'/items' profiles. Review-based users/items' profiles in the unstructured form are converted into the structured feature vectors. The structured versions are used as the input for the autoencoder. The autoencoder encodes the feature vector obtained as input in the training phase with the encoder layers and tries to reconstruct it in the output layer by decoding it with the decoder layers. When the autoencoder is trained, it is refed with the feature vectors used for training, and low-dimensional, complex and hidden features of these feature vectors are extracted from the output of the outermost encoder layer. These extracted features are used to select neighbors aiming to produce predictions in the prediction stage. In the prediction phase, AE-simMCCF++ uses the average similarity approach to calculate aggregated similarities between users/items, using both criteria-based similarities calculated from features derived from rating-based users'/items' profiles, and review-based similarities computed using the features extracted from the review-based users'/items' profiles. Overall predictions are generated using the aggregated similarities and the traditional neighborhood-based CF algorithm.

Figure 2 and Algorithm 1 show the general view and the pseudocode of the AE-simMCCF++ algorithm, respectively.

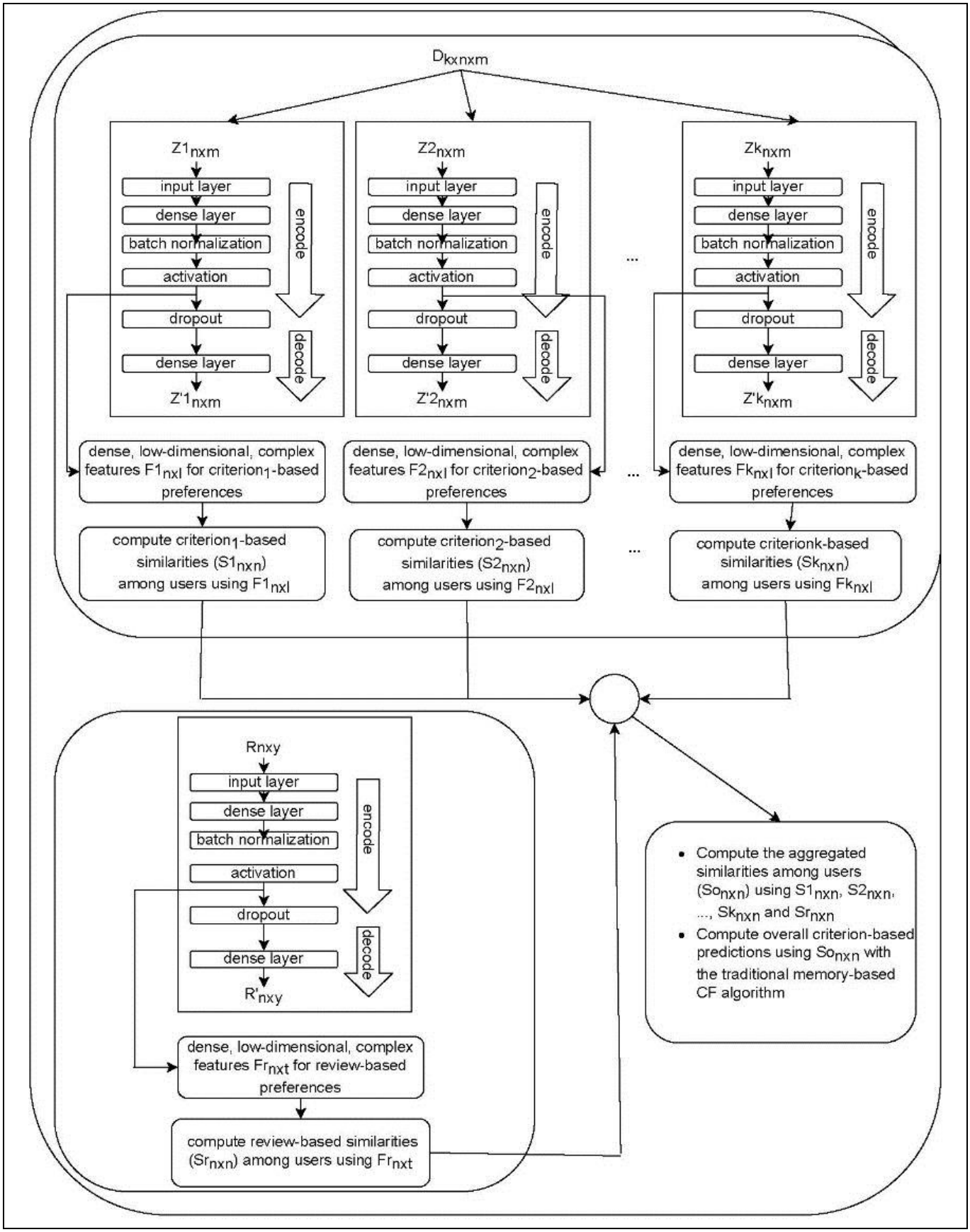

Figure 2. General representation of AEsimMCCF++.

Algorithm 1 AE-simMCCF++ method

Input: $D_{k\,x\,n\,x\,m}, R_{n\,x\,r}$ ▷ $D_{k\,x\,n\,x\,m}$ and $R_{n\,x\,r}$ are the multi-dimensional user-item rating matrix and the feature matrix obtained from review-based users' preferences,

Output: $P_{n\,x\,m}$ ▷ The produced predictions based on overall criterion,

First Stage: ▷ This stage includes feature extraction from rating-based and review-based users' preferences using autoencoders,
Feature extraction from rating-based users' preferences:
Divide D into k nxm criterion-based matrices.
For each nxm criterion-based matrix Z,
Give 0 to the missing ratings in Z,
Apply the sign function to the missing votes in Z, and record the results into the matrix Z' ▷ this step provides the network to ignore the errors caused by the missing votes,
Convert the values in Z into the ranges either [0, 1] and [-1, 1] according to the selected activation.
Create and autoencoder A for Z ▷ A includes m neurons at both input and output layers,
Network training:
Feed A with each rating-based user profile in Z,
Encode the input as described in Eq. (4),
Decode the encoded data using Eq. (5),
Compute the error value for each output neuron and multiply it with the corresponding value in Z′,
Update the bias and weight matrices for A using the chosen loss and optimization functions,
Obtain A' which is the trained version of A,
Feature extraction/Dimensionality reduction:
Feed A' with each rating-based user profile in Z,
Use the output of the innermost encoder layer as low-dimensional, complex, non-linear features obtained from raw rating-based user profiles,
Construct the criterion-based feature matrix F_{nxl} by feeding A' with each rating-based user profile in Z ▷ l is the number of neurons in the innermost encoder layer,
Extracting features from review-based users' preferences:
Map the values in the $n\,x\,r$ review-based users' preference matrix R into one of the ranges either [0, 1] or [-1, 1] according to the selected activation,
Construct an autoencoder A for R ▷ A includes r neurons at its input and output layers,
Feed A with each review-based user preference in R,
Encode the input as described in Eq. (4),
Decode the encoded data using Eq. (5),
Compute the error values,
Update the bias and weight matrices for A using the selected loss and optimization functions,
Obtain A' which is the trained version of A
Feature extraction/Dimensionality reduction:
Feed A' with each review-based user preference in R,
Use the output of the innermost encoder layer as low-dimensional, complex, non-linear features obtained from raw review-based user profiles,
Procure the feature matrix Fr_{nxt} by feeding A' with all review-based users' preferences in R ▷ t is the number of neurons in the innermost encoder layer,
Second stage: ▷ It includes constructing P,
For each nxl criterion-based feature matrix F,
Calculate the similarities among users utilizing F_{nxl} and Cosine correlation,
Calculate the similarities among users utilizing F_{nxl} and Cosine correlation,
Calculate the aggregated similarities among users So_{nxn} by applying the average similarity approach to the computed review-based similarities and rating-based similarities,
Compute P using So and traditional neighborhood-based CF algorithm.

The first stage of the AE-simMCCF++ approach includes two basic steps as extracting features from both rating-based and review-based users' preferences. For the set of users denoted by U with n users and the set of items denoted by I with m items, rating-based user-item preferences and user-item reviews are expressed as $k\,x\,n\,x\,m$ and $n\,x\,m$ matrices for a k-dimensional multi-criteria recommendation system. In the rating-based user-item preference matrix, each cell represents a user's criterion-based rating for an item. In the user-item review matrix, each cell represents a user's review of a product. For the extraction of complex, low-dimensional and dense features from the raw, high-dimensional and sparse criterion-oriented rating-based user profiles, the steps of the AE-simMCCF approach as described in the work [11] are performed.

In order to extract hidden and complex features with autoencoders from the raw user-item review matrix, firstly, the unstructured user-product review matrix is transformed into an $n\,x\,r$ dimensional feature matrix, where r is the number of features. For this purpose, all reviews made by a user are combined and an unstructured review-based user profile of that user is created. Such a profile is named as a document. The collection of all documents is called a corpus. In order to transform these documents into a structured form and express them with meaningful numerical information, preprocessing, feature extraction and feature selection processes, which are the basic steps of text mining, are applied [35].

- **Preprocessing:** Applying preprocessing to text documents has a significant positive effect on the accuracy of the classifiers in text classification [36]. In this study, the preprocessing steps of lowercase conversion, tokenization, removing stop-words such as "a", "an", and "the" and stemming for suffix stripping are applied to all documents. To examplify the review "*Ticks all the boxes Stayed at Hilton Lincoln Centre for one night 17th July Whilst on vacation and this is clearly a business hotel it is difficult to find any fault with this hotel Check in was efficient and friendly the concierge was excellent the executive lounge was fine room and other facillities.*" is converted to "*tick box stay hilton lincoln centr one night 17th juli whilst vacat thi clearli busi hotel difficult find ani fault thi hotel check wa effici friendli concierg wa excel execut loung wa fine room facil*" after preprocessing,
- **Feature extraction:** In the feature extraction step, numerical information is extracted from each document and a term weighting process is performed. Then, each document is converted into a feature vector using terms' weights. In this study, the bag-of-words (BoW) approach, where the order of the terms in the document is not considered and only the term frequencies are utilized, is used to represent

documents. Term frequency-inverse document frequency (TF-IDF) approach is used as a term weighting method in this study. TF-IDF is easy to implement and improves accuracy by giving more weight to words that are important in a document and less weight to words that are common across documents. Thus, it is one of the most basic methods frequently used in text mining. For a term t in the corpus, the TF-IDF weight of t for a document d_i is computed as the product of the t's term frequency (TF) value and the inverse document frequency (IDF) value as described in Eq. (7).

$$\text{TF} - \text{IDF}(t) = \text{TF}(t, d_i)\text{x}\log\frac{C}{d(t)}) \tag{7}$$

Where, $TF\ (t, d_i)$, C and $d(t)$ are the number of times the term t occurs in the document d_i, the number of documents containing the term t and the number of documents in the corpus, respectively.

- Feature selection: To decrease the number of features in the corpus for text mining applications, feature selection methods are applied [35]. In this study, the features whose document frequency values are greater than 0.9, which is an experimentally predefined threshold, are ignored. With this way, the corpus-specific stop words are removed.

After applying all these basic steps of text mining review-based preference matrix contains TF-IDF values. This matrix is represented as R_{nXr}. Aiming to extract low-dimensional, non-linear, complex features from review-based structured users' preferences, an autoencoder is constructed with r neurons at both its input and output layers. Each structured review-based user preference vector constructs the input of the autoencoder. The input data is encoded with Eq. (4) and decoded with Eq. (5). Aiming to prevent overfitting, dropout and regularization are applied to each encoder and decoder layers. Then, the weights and bias matrices of the autoencoder are updated using the loss function presented in Eq. (8) and the chosen optimization function.

$$\sum_{u_r \in U} \|u_r - t(h(u_r))\|_2^2 + \lambda(\|W_1)\|_2^2 + \|W_2)\|_2^2) \tag{8}$$

Where u_r is the structured review-based high-dimensional preference vector for user u. λ is the regularization parameter. After completing the training process of the autoencoder, it is fed with each structured review-based user's preference vector, and the output of the outermost encoder layer provides the low-dimensional, complex, and latent features obtained

from that profile. The review-based similarity among the users u and t, $S_r(u,t)$ is calculated as in Eq. (9) utilizing the features $F_r(u)$ and $F_r(t)$ extracted from the users' structured review-based profiles. The aggregated similarity among these two users is calculated as in Eq. (10) using k criteria-based similarities computed according to the algorithm [11] and $S_r(u,t)$. The overall criterion-based prediction of the user u for the item i is calculated as in Eq. 11.

$$S_r(u,t) = \frac{\sum_{i=1}^{i=t} F_r(u,\text{i})F_r(\text{t},\text{i})}{\sum_{i=1}^{i=t} F_r(u,i)^2 \sum_{i=1}^{i=t} F_r(t,i)^2} \tag{9}$$

$$S_o(u,\text{v}) = \frac{\sum_{c=0}^{c=k}(S_c(u,t)) + S_r(u,t)}{k+2} \tag{10}$$

$$P(u,\text{i}) = \frac{\sum_{t\in N(u)} S_o(u,\text{t})\ R(t,i)}{\sum_{t\in N(u)}|S_o(u,\text{t})\ |} \tag{11}$$

Where top-n neighbors of user u construct the set $N(u)$ and $R(t,i)$ represents the rating provided by the user t for item i. A toy example to provide a general perpective to the proposed approach is given in Figure 3.

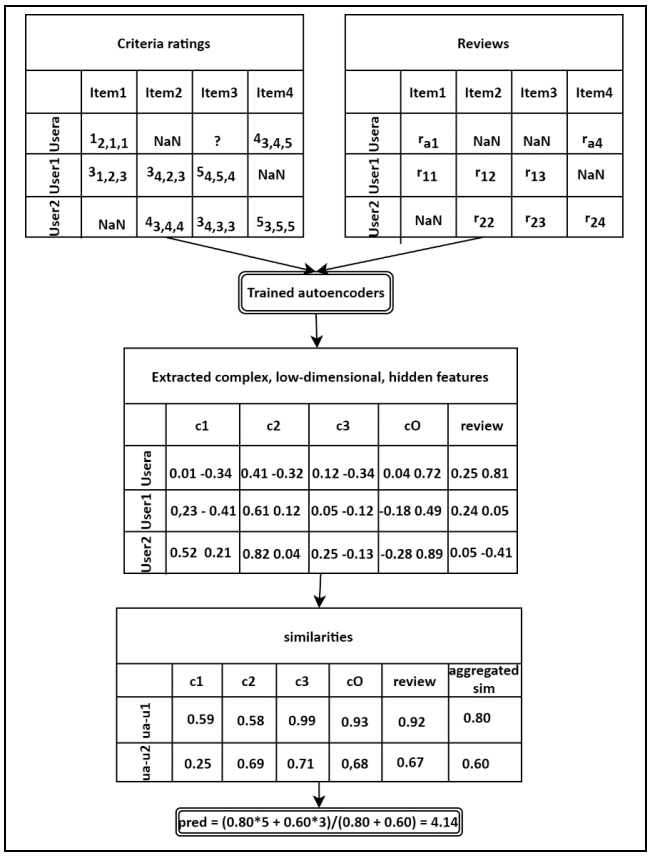

Figure 3. A toy example for AEsimMCCF++.

5 Experimental works

Several experimental studies have been carried out on a real data set in order to show the efficacy of the AE-simMCCF++ algorithm on the negative effects of the sparsity on the produced predictions in terms of accuracy and coverage. With this purpose, first of all, experimental studies have been carried out on how the varying hyper-parameters of the autoencoder such as the activation functions and the number of encoder layers affect the performance of the AE-simMCCF++ algorithm. Then, the AE-simMCCF++ algorithm is compared with the AE-simMCCF in order to show the effectiveness of using the information obtained from the reviews as well as the ratings on handling the sparsity during the prediction process. Moreover, a comparison is conducted between the proposed work AE-simMCCF++ and the state-of-the-art similarity-based MCCF algorithms. These baseline methods are listed below:

- Average similarity-based traditional MCCF method (TMCCF-AvgSim) [2],
- Minimum similarity-based traditional MCCF method (TMCCF-MinSim) [2],

- Support vector machines-based traditional MCCF approach (TMCCF-SVM) [37],
- AE-simMCCF [11].

The number of neighbors for all of the approaches is trially set to 25. Moreover, cosine similarity is used in all the experiments. Since the used experimental methodologies for the proposed work and AE-simMCCF are the same, the parameters which provide the highest accuracy and coverage values are protected for AE-simMCCF.

5.1 Data sets and evaluation measures

A subset chosen from TripAdvisor (TA) data set gathered by [38] is used in the experiments. The subset includes 4798 multi-criteria ratings collected from 1346 users for 1289 hotels. The sparsity percentage for the subset is 99.7235%. At least three ratings are provided by each user considering the criteria as value, rooms, location, cleanliness, check-in, service, and business service in addition to the overall judgments. A numeric five-star rating scale is used in the subset. In the subset, each user who has a rating vector for an item, also has a review for that item. When the reviews were transformed into a structured form, the review dataset has a sparsity of 76.3600%. This makes the review-based profiles of users to be denser than their ratings-based profiles. Aiming to construct the training and testing sets, the whole ratings in the subset are grouped as training and testing ratings in the percentage of 80% and 20%, orderly. The related reviews with the training ratings are also used as training the autoencoder, which will be used to extract features from the review-based user profiles. The reviews with regard to test ratings are used to construct the validation set that will be used in the training phase of the specified autoencoder. Each user document in the review-based training and validation set contains normalized TF-IDF values for 500 features obtained from the corpus. This procedure is repeated five times and with this way, five different rating-based train and test set pairs and review-based train and validation sets are constructed. Furthermore, due to the randomness of the hyper-parameters of the network, each analysis is repeated three times for providing reliable experiments. Thus, the result for each analysis is obtained by averaging all the outcomes of the repeated processes.

Accuracy of the produced predictions is negatively affected by sparsity. Especially for neighborhood-based MCCF approaches, finding out the most similar neighbors of a user for better accuracy becomes harder due to the absence of corated items. Moreover, producing predictions may be prevented due to the sparsity which is measured by coverage. Therefore, mean absolute error (MAE), root mean squared error (RMSE), and coverage metrics are used to present the efficacy of the proposed method regarding accuracy and coverage. Coverage is the ratio of the number of items which have predictions to all testing items [39]. Eq. (12), Eq. (13), and Eq. (14) represent the coverage, MAE, and RMSE, orderly. R_t and $|R_t|$ are the votes used for testing and the number of such votes, orderly. R_p is the set of predictions for test items.

$$Coverage = \frac{|R_p|}{|R_t|} \tag{12}$$

$$MAE = \frac{1}{|R_p|} \sum_{(i,j) \in R_p} |R_{t,ij} - R_{p,ij}| \tag{13}$$

$$RMSE = \sqrt{= \frac{1}{|R_p|} \sum_{(i,j) \in R_p} \left(R_{t,ij} - R_{p,ij}\right)^2} \tag{14}$$

5.2 Experimental outcomes

5.2.1 Impressions of the hyper-parameters

During the prediction process, the proposed method utilized the extracted features from both rating- and review-based user profiles by autoencoders. Keras 2.1.5 with TensorFlow backend is used in the trials to construct the autoencoders. Aiming to extract complex, dense and low-dimensional features from raw, sparse and high-dimensional rating-based user profiles, the autoencoder parameters with the highest accuracy and coverage values for AE-simMCCF are protected. For the autoencoder, which is designed to extract hidden and complex features from review-based user profiles, some hyper-parameters such as λ, batch size, dropout regularization and learning rate are determined experimentally and their values are preserved throughout all the experiments. λ, batch size and dropout regularization coefficient are determined as 0.001, 30 and 0.2, respectively. The weight matrices and bias vectors of the autoencoder are initialized using the He normal distribution. Adam optimizer is used as the optimizer algorithm. The default values by Keras are retained for the other parameters of the Adam optimizer, except for the decay and learning rate parameters. The decay and learning rate values are determined as 0.0001 and 0.0001, respectively.

Various experimental analyzes are conducted to show the effects of the varying activations and the number of encoder layers on the performance of the AE-simMCCF++ algorithm. For this purpose, the number of encoder layers is varied as 2, 3 and 4. The number of neurons determined for each specified layer is 1/8, 1/12, and 1/15 of the neurons in the input layer, respectively. The number of neurons used for the first encoder layer is 1/5 of the neurons in the input layer. In order to show how the varying activation functions affect the performance of the AE-simMCCF++ algorithm, nonlinear activations exponential linear unit (ELU), rectified linear unit (RELU) and hyperbolic tangent (Tanh) are used. Since vectors based on normalized TF-IDF scores contain values in the range [0, 1], these values are mapped to the appropriate ones in the range [-1, 1] for ELU and Tanh.

The effects of varying activations and the number of encoder layers on the performance of the AE-simMCCF++ algorithm are presented in Table 1. Considering the results in the table, it is concluded that the number of encoder layers that give the best results regarding accuracy and coverage varies for each activation function. This is related to the capacity of the network. Parameters such as the amount, structure and sparsity ratio of data used in training affect the capacity of the network [31]. According to the results in Table 1, the AE-simMCCF++ algorithm provides the best performance in terms of accuracy with an autoencoder that has 2 encoder layers with ELU activation. In terms of coverage, the best performance results are obtained with an autoencoder that has 3 encoder layers with Tanh activation. The most balanced performance results regarding both accuracy and coverage are provided by the autoencoder that has 2 encoder layers with Tanh activation.

Table 1. Impacts of varying activations and the number of encoder layers over the performance of AE-simMCCF++ with regards to accuracy and coverage.

Number of encoder layers	Activation function	MAE	RMSE	Coverage
2 encoder layers	RELU	0.8535	0.8559	0.1008
	ELU	**0.8427**	**0.8468**	0.1025
	Tanh	0.8441	0.8487	0.1042
3 encoder layers	RELU	0.8483	0.8508	0.1006
	ELU	0.8473	0.8512	0.1023
	Tanh	0.8449	0.8491	**0.1044**
4 encoder layers	RELU	0.8550	0.8578	0.0999
	ELU	0.8434	0.8470	0.1023
	Tanh	0.8560	0.8599	0.1025

5.2.2 Comparison with baseline algorithms

To present the efficacy of AE-simMCCF++ in terms of accuracy and coverage, a comparison between the proposed method and the baseline algorithms is conducted. Table 2 shows that integrating the complex, dense, low-dimensional features extracted by autoencoders from both reviews and criteria ratings into the prediction process can help to deal with the adverse impressions of sparsity on the produced predictions regarding accuracy and coverage. The approaches TMCCF-MinSim and TMCCFAvgSim only use the raw ratings when finding out neighbors and producing predictions which results in low accuracy and coverage values. AEsimMCCF utilizes the dense and complex features obtained from only raw rating-based user profiles in the neighbor selection phase. On the other hand, AE-simMCCF++ aims to better cope with the sparsity problem by using both the features extracted from the rating- and review-based users' profiles extracted by the autoencoders. With this way, higher accuracy and coverage outcomes are obtained compared with the other baselines.

Table 2. Comparison of AE-simMCCF++ with the baseline methods.

Algoritm	MAE	RMSE	Coverage
AE-simMCCF	0.8485	0.8525	0.1034
AE-simMCCF++	**0.8441**	**0.8487**	**0.1042**
TMCCF-MinSim	0.9382	0.9435	0.0894
TMCCF-AvgSim	0.9360	0.9412	0.0894
TMCCF-SVM	0.9348	0.9401	0.0894

Furthermore, given that the RMSE and MAE values exhibit similar trends as shown in Table 2, a statistical significance test has been conducted on the MAE metric to evaluate the improvements in accuracy provided by AE-simMCCF++., With this purpose, ANOVA test is conducted on 5 compared algorithms. According to the results given in Figure 4, Sig. value (.007) indicated that there is a significant difference between algorithms in terms of MAE at 95% confidence level. Depending on this result and the homogenous group variance, a post hoc "Tukey" test is conducted to provide pairwise comparisons between the proposed approach and the other algorithms. According to the test result presented in Figure 5, there is a significant difference between the AE-simMCCF++ and other compared algorithms except the AE-simMCCF at 95% confidence level. Descriptives presented in Figure 6 shows that the mean of the samples of AEsimMCC++ is smaller than all the other compared approaches. It is concluded that the AE-simMCCF++ is the approach that provides better accuracy results compared to the other algorithms.

ANOVA

MAE

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	.060	4	.015	4.881	.007
Within Groups	.061	20	.003		
Total	.121	24			

Figure 4. The statistical values for significancy among the algorithms.

Dependent Variable: MAE
Tukey HSD

(I) Algorithm	(J) Algorithm	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
1	2	-.0182000	.0349225	.984	-.122701	.086301
	3	-.1083800*	.0349225	.040	-.212881	-.003879
	4	-.1083800*	.0349225	.040	-.212881	-.003879
	5	-.1072760*	.0349225	.042	-.211777	-.002775

*. The mean difference is significant at the 0.05 level.

Figure 5. Statistical analysis of the algorithms in terms of MAE (1:AE-simMCCF++, 2: AE-simMCCF, 3: TMCCF-MinSim, 4: TMCCF-AvgSim, 5-TMCCF-SVM).

Descriptives

MAE

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minimum	Maximum
1	5	.822860	.0696767	.0311604	.736345	.909375	.7512	.9060
2	5	.841060	.0688999	.0308130	.755510	.926610	.7609	.9133
3	5	.931240	.0513197	.0229509	.867518	.994962	.8733	1.0091
4	5	.931240	.0513197	.0229509	.867518	.994962	.8733	1.0091
5	5	.930136	.0193730	.0086639	.906081	.954191	.9138	.9601
Total	25	.891307	.0708616	.0141723	.862057	.920557	.7512	1.0091

Figure 6. Descriptive analysis of algorithms (1:AE-simMCCF++, 2: AE-simMCCF, 3: TMCCF-MinSim, 4: TMCCF-AvgSim, 5-TMCCF-SVM).

6 Conclusions and future works

With the increasing number of criteria, the problem of sparsity becomes a more prominent problem in MCCF systems. Sparsity may cause to decline in the accuracy of the produced predictions, especially in similarity-based approaches utilizing neighbor selection. Furthermore, that problem may result in preventing to produce predictions. In this study, AE-simMCCF++, which is a similarity-based hybrid MCCF approach depending on autoencoders, is proposed to deal with the sparsity problem. AE-simMCCF++ uses the complex, dense and low-dimensional reviews' features provided by an autoencoder aiming to find out the neighbors during the prediction process. Experiments on a real data set have shown that the usage of users' reviews as well as rating-based preferences when

generating the predictions can better deal with the negative effects of the sparsity problem on the accuracy of the produced predictions.

Extracting non-linear, complex features from items' content information and users' reviews by other deep learning techniques such as convolutional neural networks and integrating them into the prediction process of MCCF systems can be considered as our future work. Additionally, analyzing performances of the proposed method and the other state-of-the-art methods in terms of beyond-accuracy metrics such as diversity is also our future work.

7 Acknowledgement

This study is supported by Eskisehir Technical University under grant 19ADP032.

8 Author contribution statements

In the scope of this study, Zeynep BATMAZ in the formation of the idea, the literature review, coding and performing the experiments and writing the draft manuscript; Cihan KALELİ in the formation of the idea, examination of the results, the spelling and checking the article in terms of content were contributed.

9 Ethics committee approval and conflict of interest statement

There is no need to obtain permission from the ethics committee for the article prepared.

There is no conflict of interest with any person/institution in the article prepared.

10 References

[1] Bulut H, Milli M. "New prediction methods for collaborative filtering". *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi,* 22(2), 123-128, 2016.

[2] Adomavicius G, Kwon Y. "New recommendation techniques for multicriteria rating systems". *IEEE Intelligent Systems,* 22(3), 48-55, 2007.

[3] Bilge A, Kaleli C. "A multi-criteria item-based collaborative filtering framework". *11th International Joint Conference on Computer Science and Software Engineering*, Chonburi, Thailand, 14-16 May 2014.

[4] Adomavicius G, Manouselis N, Kwon Y. *Multi-Criteria Recommender Systems*. Editors: Ricci F, Rokach L, Shapira B, Kantor PB. Recommender Systems Handbook, 769-803, Boston, MA, USA, Springer, 2011

[5] Altan G. "Breast cancer diagnosis using deep belief networks on ROI images". *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi* 28(2), 286-291, 2021.

[6] Cevik F, Kilimci ZH. "The evaluation of Parkinson's disease with sentiment analysis using deep learning methods and word embedding models*". Pamukkale University Journal of Engineering Sciences* 27(2), 151-161, 2021.

[7] Wang X, Kadıoğlu S. "Modeling uncertainty to improve personalized recommendations via bayesian deep learning". *International Journal of Data Science and Analytics*, 16, 191-203, 2023.

[8] Du YP, Yao CQ, Huo SH, et al. "A new item-based deep network structure using a restricted boltzmann machine for collaborative filtering". *Frontiers of Information Technology & Electronic Engineering,* 18(5), 658-666, 2017.

[9] Zhao Z, Yang Q, Lu H, Weninger T, Cai D, He X, Zhuang Y. "Social-aware movie recommendation via multimodal network learning". *IEEE Transactions on Multimedia,* 20(2), 430-440, 2018.

[10] Chen X. "The application of neural network with convolution algorithm in western music recommendation practice". *Journal of Ambient Intelligence and Humanized Computing*, 2020. https://doi.org/10.1007/s12652-020-01806-5.

[11] Batmaz Z, Kaleli C. "A new similarity-based multicriteria recommendation algorithm based on autoencoders". *Turkish Journal of Electrical Engineering & Computer Sciences,* 30, 855-870, 2022.

[12] Kumar Bokde D. Girase S. Mukhopadhyay D. "An item-based collaborative filtering using dimensionality reduction techniques on mahout framework". *4th Post Graduate Conference for Information Technology*, Sangamner, Maharashtra, India, 24-25 March 2015.

[13] Nilashi M, Bin Ibrahim O, Ithnin N, et al. "A multi-criteria recommendation system using dimensionality reduction and neuro-fuzzy techniques". *Soft Computing*, 19(11), 3173-320, 2015.

[14] Nilashi M, Esfahani MD, Roudbaraki MZ, et al. "A multi-criteria collaborative filtering recommender system using clustering and regression techniques". *Journal of Soft Computing and Decision Support Systems,* 3(5), 24-30 2016.

[15] Shambour Q, Hourani M, Fraihat S. "An item-based multi-criteria collaborative filtering algorithm for personalized recommender systems". *International Journal of Advanced Computer Science and Applications,* 7(8), 274-279, 2016.

[16] Kermany NR, Alizadeh SH. "A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques". *Electronic Commerce Research and Applications,* 21, 50-64, 2017.

[17] Fan J, Xu L. "A robust multi-criteria recommendation approach with preference-based similarity and support vector machine". *10th International Symposium on Neural Networks*, Dalian, China, 4-6 July 2013.

[18] Zhang K, Liu X, Wang W, Li J. "Multi-criteria recommender system based on social relationships and criteria preferences". *Expert Systems with Applications*, 176, 114868-114882, 2021.

[19] Batmaz Z, Yurekli A, Bilge A, Kaleli C. "A review on deep learning for recommender systems: challenges and remedies". *Artificial Intelligence Review* 52(1), 1-37, 2019.

[20] Gunawardana A, Meek C. "Tied boltzmann machines for cold start recommendations". *2nd ACM Conference on Recommender Systems*, Lausanne, Switzerland, 23-25 October 2008.

[21] Wang X, Wang Y. "Improving content-based and hybrid music recommendation using deep learning". *22nd ACM International Conference on Multimedia*, Orlando, Florida, USA, 3-7 November 2014.

[22] Zhang S, Yao L, Xu X. "Autosvd++ an efficient hybrid collaborative filtering model via contractive auto-encoders". *40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Shinjuku, Tokyo, Japan, 7-11 August 2017.

[23] Wei J, He J, Chen K, et al. "Collaborative filtering and deep learning based recommendation system for cold start items". *Expert Systems with Applications* 69, 29-39, 2017.

[24] Lee J, Lee K, Park J, et al. "Deep content-user embedding model for music recommendation". 2018. arXiv preprint arXiv:1807.06786.
[25] Seo S, Huang J, Yang H, et al. "Interpretable convolutional neural networks with dual local and global attention for review rating prediction". *11th ACM Conference on Recommender Systems*, Como, Italy, 27-31 August 2017.
[26] Paradarami TK, Bastian ND, Wightman JL. "A hybrid recommender system using artificial neural networks". *Expert Systems with Applications,* 83, 300-313, 2017.
[27] Wang Z, Xia H, Chen S, et al. "Joint representation learning with ratings and reviews for recommendation". *Neurocomputing,* 425, 181-190, 2021.
[28] Tallapally D, Sreepada RS, Patra BK, et al. "User preference learning in multi-criteria recommendations using stacked auto encoders". *12th ACM Conference on Recommender Systems*, Vancouver, BC, Canada, 2-7 October 2018.
[29] Batmaz Z, Kaleli C. "Ae-mccf: An autoencoder-based multi-criteria recommendation algorithm". *Arabian Journal for Science and Engineering,* 44(11), 9235-9247, 2019.
[30] Nassar N, Jafar A, Rahhal Y. "A novel deep multi-criteria collaborative filtering model for recommendation system". *Knowledge-Based Systems,* 187, 104811-104817 2020.
[31] Hasan, E., Ding, C., Cuzzocrea, "A. Multi-criteria rating and review based recommendation model". *2022 IEEE International Conference on Big Data,* Osaka, Japan, 17-20 December 2022.
[32] Wang, J. "Multi-criteria recommender system with hybrid deep tensor decomposition". *4th International Conference on Data Storage and Data Engineering,* Barcelona, Spain, 18-20 February 2021.
[33] Chen Z, Gai S, Wang D. "Deep tensor factorization for multi-criteria recommender systems". *2019 IEEE International Conference on Big Data*, Los Angeles, CA, USA, 9-12 December 2019.
[34] Goodfellow IJ, Bengio Y, Courville AC. *Deep Learning. Adaptive Computation and Machine Learning*, 1st ed. Massachusetts, USA, MIT Press, 2016.
[35] Uysal AK, Gunal S. "A novel probabilistic feature selection method for text classification". *Knowledge-Based Systems,* 36, 226-235, 2012.
[36] Uysal AK, Gunal S. "The impact of preprocessing on text classification". *Information Processing & Management,* 50(1), 104-112, 2014.
[37] Jannach D, Karakaya Z, Gedikli F. "Accuracy improvements for multi-criteria recommender systems". *13th ACM Conference on Electronic Commerce*, Valencia, Spain, 4-8 June 2012.
[38] Wang H, Lu Y, Zhai C. "Latent aspect rating analysis without aspect keyword supervision". *17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, USA, 21-24 August 2011.
[39] Ge M, Delgado-Battenfeld C, Jannach D. "Beyond accuracy: evaluating recommender systems by coverage and serendipity". *2010 ACM Conference on Recommender Systems*, Barcelona, Spain, 26-30 September 2010.