



# A deep learning-based hybrid model for recommendation generation and ranking

N. Sivaramakrishnan<sup>1</sup> · V. Subramaniaswamy<sup>1</sup> · Amelec Viloría<sup>2</sup> · V. Vijayakumar<sup>3</sup> · N. Senthilselvan<sup>1</sup>

Received: 5 December 2019 / Accepted: 6 March 2020 / Published online: 2 April 2020  
© Springer-Verlag London Ltd., part of Springer Nature 2020

## Abstract

A recommender system plays a vital role in information filtering and retrieval, and its application is omnipresent in many domains. There are some drawbacks such as the cold-start and the data sparsity problems which affect the performance of the recommender model. Various studies help with drastically improving the performance of recommender systems via unique methods, such as the traditional way of performing matrix factorization (MF) and also applying deep learning (DL) techniques in recent years. By using DL in the recommender system, we can overcome the difficulties of collaborative filtering. DL now focuses mainly on modeling content descriptions, but those models ignore the main factor of user–item interaction. In the proposed hybrid Bayesian stacked auto-denoising encoder (HBSADE) model, it recognizes the latent interests of the user and analyzes contextual reviews that are performed through the MF method. The objective of the model is to identify the user’s point of interest, recommending products/services based on the user’s latent interests. The proposed two-stage novel hybrid deep learning-based collaborative filtering method explores the user’s point of interest, captures the communications between items and users and provides better recommendations in a personalized way. We used a multilayer neural network to manipulate the nonlinearities between the user and item communication from data. Experiments were to prove that our HBSADE outperforms existing methodologies over Amazon-b and Book-Crossing datasets.

**Keywords** Deep learning · Optimization · Side information · Hybrid model · Recommendation system · Collaborative filtering

---

✉ V. Subramaniaswamy  
swamy@cse.sastra.edu

N. Sivaramakrishnan  
nsk@cse.sastra.edu

Amelec Viloría  
aviloria7@cuc.edu.co

V. Vijayakumar  
vijayakumar.varadarajan@gmail.com

N. Senthilselvan  
senthilselvan@cse.sastra.edu

- <sup>1</sup> School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India
- <sup>2</sup> Departamento de Gestión Industrial, Agroindustrial y de Operaciones, Universidad de la Costa CUC, Barranquilla, Colombia
- <sup>3</sup> School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

## 1 Introduction

The available information is sharply increasing with an online platform developed in recent times. The recommender system is significantly playing a vital role in storing massive information. RS intelligently captures the content for the user to enable easy navigation based on their past preference. To assist the users in searching items, they are looking for; RS intends to extract the required information. Hu et al. [1, 2] modeled RS with item-oriented information which can appropriately reinforce the manufacturers’ yield. In Zhang et al. [3], the author has provided an RS model to forecast implicit drugs’ side effects screens improper drugs for users, which can gallantly assist in improving the effectiveness of the medical treatment. Among all the RS models, the CF-based RS is the most

prosperous recommendation method, which creates the representation established on rating information between the user and item features [4]. The prominent way of all CF established methods is matrix factorization [5], which is used as a powerful tool for rating prediction. The user–item synergy-related CF method has evolved and been performed using matrix factorization (MF), which became the common approach adopted after the Netflix Prize competition. There are many methods of recommender systems extensively applying MF at present, which shows the improved achievement of RS through distinctive facets such as interest exploring and the community of social environment. The traditional CF models [5, 6] straightaway take up the vector-based rating information to generate the representation between the user and item features; using MF-based models, these features can be designed by low-dimensional space of latent factors.

Information about rating plays a significant role in CF models [7]. Implicit and explicit types are used for rating-based RS [8]. The binary form of the score is supplied in the implicit rating. With the help of explicit rating, a multi-value-based scoring value is provided. These two types of rating information can provide knowledge in terms of sentimental and semantic when providing representation between the user and item feature. Due to data sparsity problem [9], user can have communication with the small number of items. For solving this problem, side information is comprehensively applied to enhance better data resources of the RS. The side information commonly provides textual information about an item, user’s review about an item, etc. It offers complete knowledge about user and item; it includes semantic and sentiment knowledge about the user. Result of that leads to the generation of noise to the RS model. Luckily, deep learning (DL) models work in a better way to handle these noise data. Salakhutdinov et al. [10] and Wang et al. [11] focus on combining RS models with DL methods. Among the various DL methods, stacked denoising auto-encoder (SDAE) [12] is preferred by the many due to its positive generation of result in particular with extracting information from textual data. SDAE has been applied to select the various content features of items, which is validated by comparing it with different traditional methods. For training RS, both the implicit ratings and the explicit ratings are used. In most of the conventional approaches, the side information is not adequately applied to generate effective results.

In this paper, to handle three kinds of information, namely implicit rating, explicit rating and side information, we proposed a novel classification model called hybrid Bayesian stacked denoising auto-encoder (HBSADE), which is incorporated with MF representation. In the following sections, we introduce the sub-models. One sub-model is used for combining the explicit rating and side

information, whereas the other sub-model is integrating the implicit rating and side information.

This article’s significant contributions are listed as follows:

- Proposed recommendation system is used to explore the point-of-latent-interest distribution of the users’ through sparse latent Dirichlet allocation (sparse LDA) received from the textual review. Accordingly, this system generates personalized recommendations from learned interest.
- Through improving the prediction accuracy, the proposed system obtains factors for both the textual and contextual review information for items with the help of a conditional neural network.
- The proposed system uses neural generalized matrix factorization (NGMF) to determine low-rank characteristic vector values for both the users and the items.
- By applying stochastic gradient descent (SGD), the optimized list of candidates is generated with the exponential growth of local minima.
- A three-layer stack-based denoising auto-encoder (SDAE) model is utilized to rank top-N recommendation by taking into account different information.
- The proposed method uses a novel hybrid approach that recognizes the latent interests of the user and analyzes contextual reviews. It outperforms existing methodologies PMF, CDL and CMF over Amazon-b and Book-Crossing datasets.

The proposed system can work with unbalanced datasets and deals with explicit and implicit feedback studies that show the proposed method gives better results in comparison with recommendation approaches, specifically in recommendation accuracy and efficiency. The proposed method provided an implicit kind of feedback system where we will get user–item interaction. And the proposed work uses candidate ranking by analyzing side information.

The remaining portion of this article is formulated as follows: We present the related work about deep learning-based recommendation systems in Sect. 2 and the background details of this work in Sect. 3. In Sect. 4, we offer the problem definition and general framework. Section 5 describes the proposed work. Section 6 reports the experimental evaluation on two real-world datasets Amazon-b<sup>1</sup> and Amazon-m&t, and the conclusion part and the future work are presented in Sect. 7.

<sup>1</sup> <http://jmcauley.ucsd.edu/data/amazon/>.

## 2 Related works

Previous models make use of data from explicit feedback, which is used as a primary source for recommendation tasks. Still, it is slowly moving in the direction of the implicit form of data—CF’s implicit feedback considered as a recommendation problem that stresses implementing a simple item list for users [23]. Finding a rating problem is determined by the task compassed on explicit kind of feedback (EF), but this is almost pragmatic to decide on the recommendation items. Still, it also is considered a demanding and tedious task. Implicit feedback followed for recommending items. Implementation plan is prepared with two feedback strategies that examine the missing data and also includes a weight measurement process. Whenever we have a missing data problem, devoted representations have been put forward by He et al. [24]. Rendle et al. [25] implemented implicit coordinate descent (iCD) for the representations based on feature-based factorization that successfully reached cutting margin achievement to item recommendation. The neural network convention for the recommendation system is explained in the following descriptions.

It has been believed that the generated recommendation confers to the user’s interest that enhances RS’s performance. It is intractable to decipher the interests of each user. Researchers have used the learning of transfer to acquire the main interests of users, due to efficiency with latent Dirichlet allocation (LDA). The modeling of “document-topic-words” is similar to “user-interests-items.” This method is used to improve the accuracy of recommendation with “interest exploring” via LDA [14]. Wang et al. proposed probabilistic modeling of a topic through LDA to use topics similar to users under document recommendation preference. The point-of-interest recommendation is supported by Ren [27]. Experimental analysis on domain datasets shows that the interest recommendation approaches outperform the existing approaches.

Probabilistic matrix factorization (PMF) is employed to extract knowledge about latent features and carry out a prediction for rating through the product of extracted feature vectors. Existing recommendation methods are on a new version of the popular PMF. It manages limitations such as data sparsity as well as measuring linearly with an equal amount of considerations. Social networks, contextual and other information are used to improve prediction accuracy. To improve performance, PMF and DL such as CNN and auto-encoders (AEs) have been combined recently [28] into NGMF. In this paper, NGMF is further engaged as a fundamental component for low-rank feature/characteristic vectors.

DL methods are employed in the proposed work to improve the performance of recommendations, including

CNN and SDAE. The work performed by Salakhutdinov et al. [15] includes a two-layered restricted Boltzmann machine for representing explicit ratings for user items [30]. This work was used to describe ratings for ordinal nature. At present, the most commonly applied option to construct a recommendation system is via auto-encoders [17]. It focuses on the “hidden patterns” study that can reconstruct the user’s rating with inputs of historical evaluation, termed as user-based AutoRec [29]. The choice of user data personalization in this method shares similarity information with the user–item representation, where all rated items indicate the users’ preferences. The objective of eliminating auto-encoders is to find function learning and provide failure results to generalize the sensed data. The characteristics obtained as a result of the deep neural network further integrated with MF. The one which resembles this work ensures the auto-encoder of collaborative filtering, named as collaborative filtering denoising auto-encoders (CDAEs) [18] with the representation of implicit kind of feedback (IF).

In contrast to the denoising auto-encoder-based collaborative filtering, the CDAE advances one node of corresponding-user auto-encoders input for the reformation of users’ ratings. As reported by authors, CDAE displays some standard features with singular value decomposition (SVD) representation, where the application of identity function can be used to obtain the hidden structures’ activation of CDAE. Even though CDAE uses the method of neural representation for collaborative filtering, it further considers the inner-product value to represent the user and item interactions (UII). The application of deep learning layers for collaborative filtering auto-encoders does not increase its performing ability. Due to the stereotypical behavior of collaborative denoising auto-encoders, this shows a two-way hierarchical model where the item and user communications are prototyped with multilayer organized in the form of a feed-forward neural network model. It helps to assess an arbitrary function from data given, which is quite self-explanatory, as well as having more capability than the actual inner-product function (IPF), which produces a constant value. Similarly, in the previous works of knowledge-based graphs, the communication between the two different objects has been rigorously worked out. There has been a lot of development in the machine.

## 3 Preliminaries

A recommender system intelligently captures the content for the user to enable easy navigation based on their past preference. In recent years, the quantum of online information has increased massively and therefore finding

useful information becomes a severe problem. It takes the information overload and accomplished good results in bountiful industries. RS provides the leverage of 80% selection of movies available on Netflix and 60% of movies on YouTube. However, RS lacks abundant knowledge on users' innate interest, yielding poor performance in results. In the current developing world, users merely make any decision over items based on their primary interest and performance of the product. This kind of recommendation system provides various valid suggestible approaches that could give benefits to the end user.

### 3.1 The traditional way of generating recommendations

RS is modeled at the beginning stage to anticipate the ratings of missing values and to generate top-N recommendations for challenges, concerning the past behavioral records. The prominent method of all CF established methods is MF [13], which is useful as a powerful tool for rating prediction. User-item synergy-related CF method modeling has evolved and been performed using MF, which became the default after the Netflix Prize competition. There are many methods of recommender systems extensively applying MF at present, which shows the improved achievement of RS through distinctive facets such as interest exploring and the community of social environment. Research is undertaken to develop MF, and neighbor-based representation has been integrated with MF; topic representation of item description also has been added, promoting the functional capabilities of MF. The selection of interaction function has a negative impact, though it is sufficient for collaborative filtering. Explicit feedback will enhance the performance of MF representation. Communication established between the products and the users called latent features can be designed or modeled with minor changes applied to the inner-item operator. This inner-item operator joins features of the product in a linear type and is not sufficient for getting a complex representation of the data about user communication. Moreover, added information is converted into normalized values as regularization parameters to force MF to acquire knowledge about low-rank characteristic points for the users and the items.

The problem must address existing collaborative filtering methods based on implicit data, followed by a well-known technique called MF, and its restriction due to latent inner-level user products and the item vector values [26]. In real life, the accuracy of the recommender system is very low, but it is one of the demanding tasks for RS to figure out the cold-start and the data sparsity problems, as well as an unsecured recommendation with a high form of accuracy. The goal is to provide top-N recommendations

generated as a list for the set of queries that are not performing the prediction of ratings. For unbalanced datasets, final recommendation results and the performance of RS are not stable. On the other hand, available information is consistently neglected by many researchers, the use of which helps to improve the accuracy of recommendations.

### 3.2 DL-based recommendations

DL techniques have been developed for application to any real-world application fields including speech recognition, image classification, text processing, sentiment analysis, etc. Many studies have sought to introduce the latter into the area of RS to improve the performance in comparison with traditional RS. In contrast to conventional RS, a vast number of researchers seek to introduce a way of deep learning models, which is to improve the performance of RS. Salakhutdinov et al. [15] incorporated restricted Boltzmann machines (RBMs) into collaborative filtering, which contains a hidden and visible layer. A multilayer perceptron (MLP)-based recommender engine utilizing information from different sources is given for YouTube, with different hidden layers between the input and the output layers. Gao [16] employs MLP for document recommendation.

Auto-encoders are used for recommendation, whose objective is to reform the ratings of input in the output layer [28]. Kim [18] introduced the convolutional neural network (CNN) into MF for document recommendation to utilize text data. Zhang [19] conducted a study of deep learning-based recommendation systems, which is helpful to future researchers. The integration of deep learning techniques with traditional recommender systems is depicted in Fig. 1.

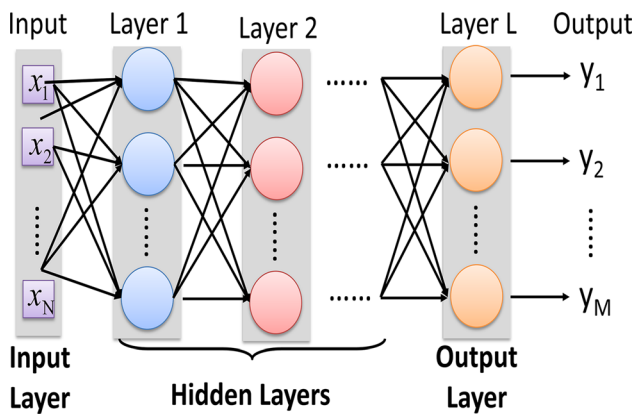
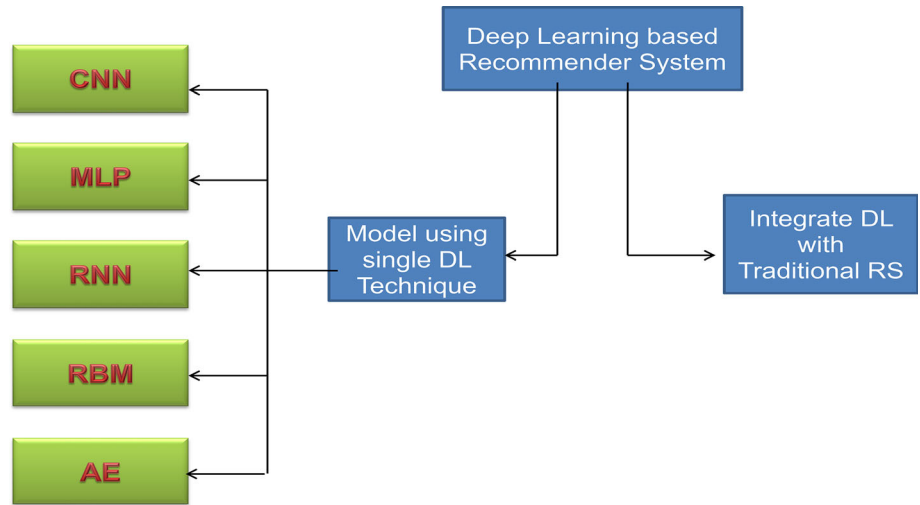
Recommender systems that are incorporated by deep learning models are producing vast data (available) with different attributes. However, it is shown that deep learning representations acquire complicated information with the numerical and textual data, which works well with unbalanced data and provides better yield performance.

### 3.3 Deep learning and artificial neural network

DL, being robust in the machine learning family, acquires knowledge about data representation instead of a task-specific algorithm. DL models use neurons, which are a deluge of multilayered nonlinear processing units [25] used to accomplish the extraction of features and manipulation that are preprogrammed [20, 21]. An environmental representation of neurons mentioned above is referred to as an artificial network.

The information processing capability of a biological neural network available in the human brain has inspired

**Fig. 1** Integration of deep learning techniques with traditional recommender system



**Fig. 2** Neural network with multiple combinations of layers

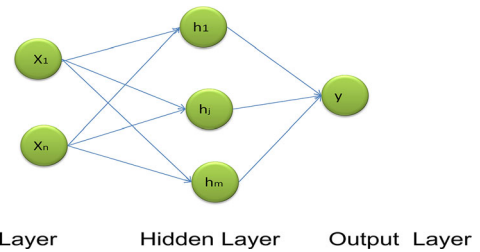
the computational model that is an artificial neural network (ANN). The computation unit—neuron, is often referred to as a node [36]. It collects input from another set of neurons and measures the combination of output—the organization of the neural network depicted in Fig. 2.

### 3.4 Feed-forward neural network

A neural network is an organized form of a system utilizing neurons arranged in layers that have connections and weights associated with the neurons of adjacent layers. The example of a feed-forward neural network is shown in Fig. 3. There are three different nodes in a feed-forward neural network:

*Input nodes* The input node does not perform any computation but shifts the information from the external world to the system. The layers with input nodes referred to as input layers.

*Hidden nodes* The hidden node does computations and shifts the information to output nodes, but it has no



**Fig. 3** Feed-forward neural network

connection with the external world. This layer is referred to as an invisible layer.

*Output nodes* The output node does the computations and shifts the information to the external world. This layer has output nodes.

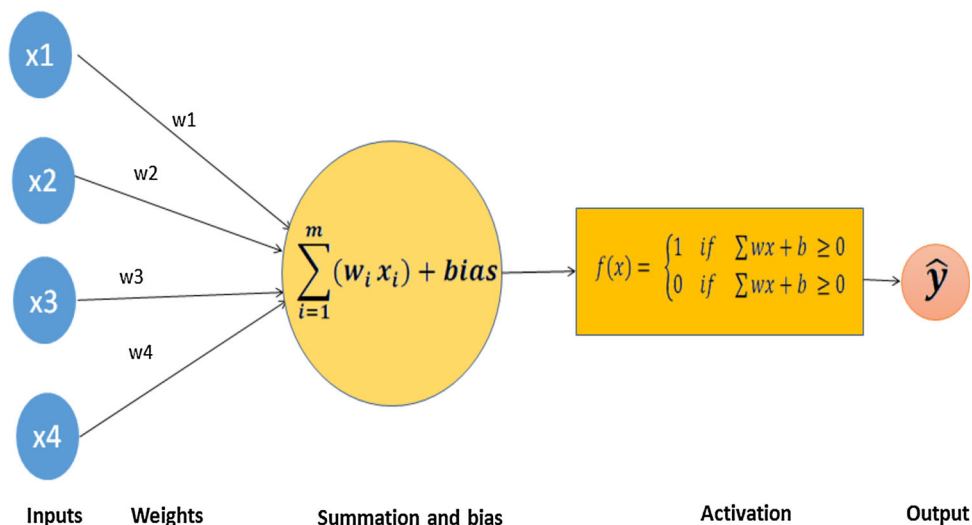
In the feed-forward network model, the flow of information is only in a straight line, particularly in one direction and not in any cycles or loops. There are two types of feed-forward networks.

- Single-layer perceptrons—there is no hidden layer in this form of network.
- Multilayer perceptrons—the network comprises one or more hidden layers.

### 3.5 Multilayer perceptron

Two different pathways are applied to represent user and item, which is made up of neural collaborative filtering. This form of network joins both highways to customize a profound learning recommender system. Mere vector interaction is not satisfying to see user–item interaction. To resolve this issue, MLP is used to study the interaction between user and module inert vectors [22, 37], and we

**Fig. 4** Multilayer perceptron-based neural network



applied masked surfaces on the concatenated vector. ReLu [23] is used as an activation function to develop the architecture, and the tower pattern is used to represent a neural network architecture in which the base is the extensive one. Every continuous layer has sub-units of the neuron. The association weight of each input to the node conveys its relative emphasis to the other set of data. The original function applied by the node to the weighted form of a sum of inputs is displayed in Fig. 4.

The summation is calculated together with bias value. The activation function ( $f$ ) is in nonlinear form, and it is helpful in understanding complex patterns in data. Finally, it produces a knowledgeable kind of information. By using deep learning algorithms, the objective function is used to calculate the model parameters. There are two different methods used, namely point-wise loss objective function and pair-wise loss objective function. In the point-wise loss objective function, they follow this model also handles a registration process. They are considered either by sampling the negative entries or by considering all the unidentified entries as negative feedback. The identified entries are ranked higher than the anonymous entries.

#### 4 Problem definition and general framework

In this section, a detailed study is provided on the exact problem definition and general framework model that consists of hybrid deep learning-based collaborative filtering, including sparse LDA, NGMF, MLP and stacked Bayesian denoising auto-encoders (SBDAEs).

#### 4.1 Problem definition

The use of this recommendation system is to provide a timely recommendation to users. It was a challenging task for previous researches to solve this kind of problem in the past. DL methods generate an accurate solution for a recommender system. The responsibility of the proposed system is to provide a comprehensive list of top-N recommendations. First, each user's interest exploration has been shown as output representation. In the past, many researchers explored the user's interest in various courses of action. LDA could be applicable for discovering topics among a group of distinct words.

It inspired deep learning-based machine recommendation (DLMR), a standard scheme of three layers proposed for interest extraction from the available database of textual information, which includes reviews that could mirror the user's interest and preference. The resorting of representation of sparse LDA is performed to mimic interest interference tasks. Users always perform decision making concerning their linkings and interests in real-world life. It is believed that the user-oriented machine has robustness. In fact, in real-world experience, the user's original attitude may not match, to some extent, with the interests you have learned. By using another option, the naive statistics of past behavior generate another solution that approximates distributions of user benefits. Additionally, it introduces the interest distribution for users in this way.

Next, the interest coefficient obtained a score: the approximated degree value among the distribution of the interest known by applying sparse LDA and the initial interest of distribution whose value ranges from 0 to 1. Additionally, it was incorporated into NGMF, which acts as a regularization term to limit feature vector learning.

Traditional matrix factorization has been a popular technique to handle recommender system problems. In this method, the user–item communication is merged with a real-world vector of latent features. The latent space is referred to as  $k$ . The interaction between the different users and the latent product factors is by seeing every flow of available latent space, which is not adequately interconnected with each other, and they are linear with a similar set of workloads. Hence, MF is a 1-D representation of latent factors. Two settings have to be stated clearly beforehand. The first setting is the dot product of the cosine value of the angle in the latent vector, which provides the idea about everyday things between two different people. The second phase is performing the Jaccard coefficient similarity between the users and items.

From Fig. 5, we can infer that user–item matrix (a) u4 is quite identical with user–item matrix u1, followed by u3 and then u2. In user latent space (b), p4 is near to p2 than p3. It helps to measure the identical activities between the users without laying back the standard behavior. On the other hand, u1 follows u3, which is followed by u2. But, p4 is kept closer to p2 than p3, but it has resulted in a more significant ranking loss. Because of this problem, we preferably proceed with a deep learning method called NGMF.

### 4.2 Neural generalized matrix factorization (NGMF)

Using this model, NGMF is interpreted as neural collaborative filtering as a particular case. The large family of factorization is covered by modeling to NGMF. One hot-encoding model is the input of the user/item vector and embedding layer as a latent model of vector value combination of user/item. Consider  $p_u$  as user latent model vector combination, and  $q_i$  is item latent vector value. We need to specify the mapping function to the very first neural representation of CF layer as

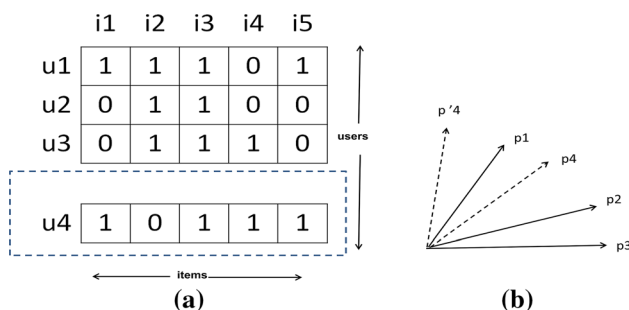


Fig. 5 a, b MF’s drawback example with user–item matrix and user latent space

$$\vartheta_{out}(p_u, q_i) = p_u \odot q_i \tag{1}$$

where  $\odot$  represents the dot product value of two vectors. Again, we apply vector projection to the output layer as

$$\hat{y}_{ui} = a_{out}(h^T(p_u \odot q_i)) \tag{2}$$

where  $a_{out}$  and  $h^T$  both are used as activation function and corresponding output layer’s edge weights. Sigmoid function as activation function is used in the generalized version of MF and model parameters known with log loss objective function.

Until now, we have gone through the neural network-based architectures—NGMF, which uses the linear model of the kernel function, and MLP, which uses a kind of nonlinear kernel, jointly, to study communication methods from data. To absorb the complicated user–module interactions, we show a hybrid architecture by combining MLP and NGMF, so that they can mix and interact with each other. An evident method to combine these architectures is to share MLP and NGMF standard embedding surface and further integrate the outputs of their actual interaction functions. Nevertheless, the performance flexibility of combined architecture decreased while sharing embeddings of NGMF and MLP. Thus, in order to study distinct embeddings and to integrate these architectures through concatenating their final masked surfaces, as shown in Fig. 6, we allowed MLP and NGMF.

By combining NGMF and MLP, we can limit the achievement of the fused representation, so we gave permission to perform a combination of these two. We formulate this representation as:

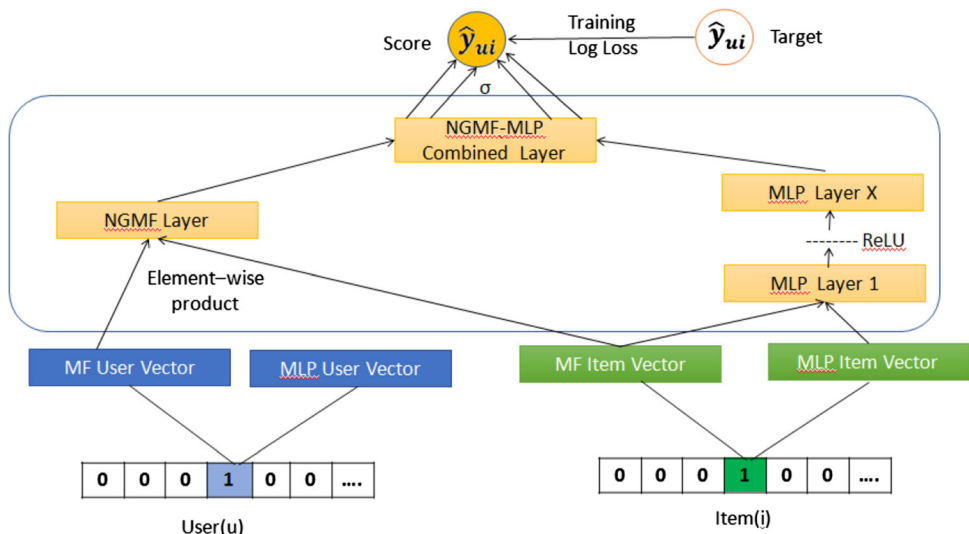
$$\hat{y}_{ui} = \sigma(h^T(\vartheta_{out}^{NGMF} \cdot \vartheta_{out}^{MLP})) \tag{3}$$

where NGMF is the most distinct method under collaborative filtering research, and user–item communications are recognized as a constant inner-product user–item input matrix. MLP is used to learn the input–output combinations. This representation combines linear and nonlinear neural network-based MF for designing user–item latent form of structures.

### 4.3 A hybrid deep learning-based collaborative filtering model

In most of the collaborative filtering-based recommender systems, it is very difficult to infer latent factors for both the users and items from the given raw inputs. Implicit kinds of relationships between users and items are only captured using MF-based collaborative filtering recommender systems. Additionally, they face problems called data sparsity and cold-start problems. Moreover, deep learning neural network models have been shown to be highly effective in identifying high-level hidden models

**Fig. 6** Embedded form of neural generalized matrix factorization (NGMF)



from the original input for a variety of tasks. So, there is an urge to make use of deep learning neural network models effectively to improve the performance of collaborative filtering.

In this section, we propose a hybrid deep learning-based collaborative filtering model which integrates the functionalities of Bayesian stacked auto-denoising encoder (BSADE) and NGMF-based collaborative filtering-based recommender systems. The proposed hybrid model makes use of both the rating matrix and side information, which combines BSADE and neural generalized matrix factorization. Neural generalized matrix factorization models are best suited for handling problems such as scalability and accuracy. On the other hand, BSADE is very powerful in managing the massive volume of raw inputs and extracts a high-level model from these inputs [31]. The combination of these two models outperforms the recommender system in a better way. The BSADE stacks various DAEs together to create a high-level model. The model of BSADE is represented in Fig. 7, and the deep learning design model comprises various steps which are listed as follows:

For every hidden layer  $l \in \{1, 2, \dots, L - 1\}$  of the BSADE representation (as represented in Fig. 7), the hidden model  $h_l$  is measured as:

$$h_l = g(W_l h_{l-1} + V_l \tilde{x} + b_l) \tag{4}$$

where  $h_0 = \tilde{s}$  is cone among the corrupted inputs. For the output layer  $L$ , the final outputs are produced as:

$$\hat{s} = f(W_L h_L + b_s) \tag{5}$$

Note that the first half of the layer acts as an encoder and the second half of the layer acts as the decoder. The BSADE makes use of a deep neural network to reform the inputs and to minimize the squared loss between their inputs and their associated instructions. Correspondingly,

using backpropagation algorithm we can learn the parameters  $W_l, V_l, b_l$  for each and every layer. Latent factor vector is created for half of the layer.

$$\hat{x} = f(W_L h_L + b_x) \tag{6}$$

#### 4.4 Hybrid Bayesian stacked auto-denoising encoder (HBSADE)

The proposed model, called HBSADE, combines PMF and stacked denoising auto-encoder (SDAE), where the purpose of using deep learning techniques is to make powerful features for content information. Using a collaborative deep learning model, we can collect the feedback from rating information. It is a combined model of collaborative filtering and learning process. The collaborative deep leaning is done to complete a low-rank matrix.

- Initially, in this model, add noise to input and make the model more robust.
- Objective function is:

$$\min_{\{w_l, b_l\}} \|x_c - x_L\|_F^2 + \lambda \sum b_l \|W_l\|_F^2 \tag{7}$$

- The target for the HBSADE is that to minimize the error rate and maximize the posterior probability.

$$\arg \min (f_\theta(x) - y)^2 \rightarrow \arg \max (p(\theta|D)) \tag{8}$$

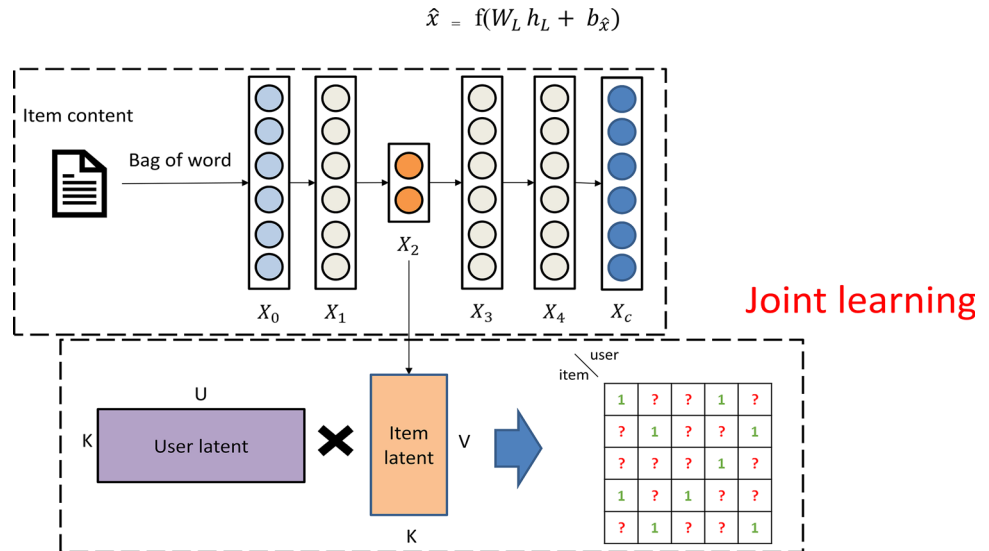
where  $(p(\theta|D))$  is calculated as follows:

$$(p(\theta|D)) = \frac{p(\theta|D), p(\theta)}{p(D)} \tag{9}$$

The step-by-step process of HBSADE is given as follows:



**Fig. 7** Bayesian stacked denoising auto-encoder



1. For each and every layer of the HBSADE architecture,

a. For each attribute  $n$  of the original weight matrix  $W$ , generate

$$W_{l,*n} \sim N(0, \mathfrak{I}_w^{-1} I_{K_l}) \quad (10)$$

b. Generate the bias vector

$$b_l \sim N(0, \mathfrak{I}_w^{-1} I_{K_l}) \quad (11)$$

c. For each and every row  $j$  of  $X_l$ , generate

$$X_{l,j*} \sim N(\sigma(X_{l-1,j*}, W_l) + b_l), \mathfrak{I}_s^{-1} I_{K_l} \quad (12)$$

2. For each and every item  $j$ ,

a. Generate a clean input

$$X_{c,j*} \sim N(X_{L,j*}, \mathfrak{I}_n^{-1} I_J). \quad (13)$$

b. Generate an offset vector for latent item  $\varepsilon_j \sim N(0, \mathfrak{I}_v^{-1} I_J)$  and assign the latent vector to be:

$$v_j = \varepsilon_j + X_{\frac{L}{2},j*}^T \quad (14)$$

3. Generate a latent user vector for each and every user  $i$ :

$$u_i \sim N(0, \mathfrak{I}_u^{-1} I_K). \quad (15)$$

4. Generate a rating matrix for each user–item pair  $(i, j)$ :

$$R_{i,j} \sim N(u_i^T v_j, C_{ij}^{-1}) \quad (16)$$

where  $\mathfrak{I}_w, \mathfrak{I}_n, \mathfrak{I}_u, \mathfrak{I}_s$  and  $\mathfrak{I}_v$  are the hyperparameters and  $C_{ij}$  is the value of confidence parameter. We have to understand that middle layer  $X_{L/2}$  actually serves as an

interface between the ratings and the content information. The middle layer, along with the offset value of the latent feature  $\varepsilon_j$ , is the main key that allows us to do the learning of the feature representation effectively and finds the similarity between items and users. We can take computational efficiency  $\lambda_s$  to infinity. HBSADE is a combined learning environment that actually learns content information by integrating SDAE and collaborative filtering for the rating matrix. HBSADE is a novel hierarchical Bayesian model working to establish the link between deep learning and recommender system. HBSADE provides such a framework, where we can change BSADE to other deep learning model or add additional information.

The proposed model comprises three major components, namely upper, middle and lower components. The upper and lower parts are responsible for the extraction of latent factor vectors, whereas the middle part decays the rating matrix  $R$  into the two latent factor matrices. The intermediate layers are responsible for capturing the similarity and relationship between the users and items.

HBSADE combines the encoder and decoder parts. The encoder  $g(\cdot)$  receives the input  $s$  and represents it to a hidden model of  $g(s)$ ; on the other hand, decoder  $f(\cdot)$  represents the hidden model back to the reformed version of  $s$ , such that  $f(g(s)) \approx s$ . The arguments of the encoder are trained to minimize the error of reformation, which is actually calculated by some loss  $L(s, f(g(s)))$ . However, HBSADE includes a slight update to the original setup. It reforms the input  $s$  from a change or debate by making errors or unintentional alterations with the training level motivational representation from the input. HBSADE is trained to reform the input  $s$  from its corrupted copy  $\tilde{s}$  by means of minimizing  $L(s, f(g(\tilde{s})))$ . Generally, the ability is

to choose corruption which combines with additive isotropic binary noise. Furthermore, various kinds of auto-encoders have been introduced in many domains to display encouraging results.

In this work, we are integrating inputs together with auxiliary side information. Assuming that the sample set  $s = [s_1, s_2, \dots, s_n]$  and the respective side information set  $x = [x_1, x_2, \dots, x_m]$ , HBSADE examines random misrepresentations over  $s$  and  $x$  acquiring  $\tilde{s}$  and  $\tilde{x}$ . It combines both the encoders and decoders by implementing the following equations (Eqs. 17–19).

$$h = g(W_1\tilde{s} + V_1\tilde{x} + b_h) \tag{17}$$

$$\hat{s} = f(W_2h + b_{\hat{s}}) \tag{18}$$

$$\hat{x} = f(V_2h + b_{\hat{x}}) \tag{19}$$

where  $\tilde{s}$  and  $\tilde{x}$  are the corrupted form of  $s$  and  $x$ , respectively,  $\hat{s}$  and  $\hat{x}$  are the reformations of  $s$  and  $x$ , respectively, and  $h$  is the hidden innate representation of the original inputs.  $W$  and  $V$  are the weight matrices,  $b$  is the bias vector, and  $g(\cdot)$  and  $f(\cdot)$  are the kernel activation functions.

In the phase of distance belief model training, using mini-batch SGD, we sought the optimized result. The model parameters are shared by partition. This allows 10 s, 100 s and 1000 s of cores per model. Gradient descent is a way to minimize an activation function  $J(\theta)$ .  $\theta \in R^d$  is the parameters and  $n$  is the learning rate.  $\nabla_{\theta}J(\theta)$  is the gradient of the activation function with regard to the arguments. Usually, parameters in the opposite direction of the gradient are updated successfully.

The updated equation is given as follows:

$$\theta = \theta - n\nabla_{\theta}J(\theta) \tag{20}$$

The computation of gradient for the entire dataset has been carried out successfully with regular updates. SGD shows the same convergence behavior as batch gradient descent if the learning rate slowly decreased (annealed) over time. The number of possible local minima grows exponentially with the number of parameters, as depicted in Fig. 8.

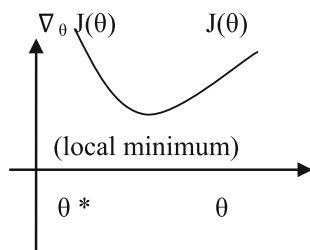


Fig. 8 Optimization with gradient descent

### 4.5 Candidate ranking

Along with the high efficiency of HBSADE presented, items receiving prediction scores with high value would be recommended to the users. Ultimately, users only may be interested only in the initial phase of HBSADE. The measurement of candidate items delivered to the raw-item database reduces significantly from millions to hundreds. HBSADE’s focus quality for prediction of the rating was never quite enough to receive satisfactory results. In this phase, the ranking of candidates plays a crucial, unique role for the definitive list of top-N recommended list of items, which will affect the achievement of the recommender system in some aspects.

In the second phase, BSADE or user-defined model of denoising encoder network model with well-known sigmoid activation function is applied for ranking of candidates in HBSADE, which is quite different from traditional ranking methods. The three hidden layers, BSADE, own the leadership overrepresentation or model and manipulation. It is flexible for BSADE to leverage available heterogeneous information for better performance. Side information (SI) includes the user’s profile with items such as time and venue. It is challenging work for the rich side to incorporate deep learning work.

The ultimate aim of BSADE is to re-rank the candidates’ lists with the available side information and provide the best top-N-generated recommendation results. BSADE could perform the re-ranking process of the candidate by taking into account accessible side information not used in the existing researches. Concerning the results obtained in DAE, HBSADE is one that provides a dozen final recommendation items available with a high score, otherwise termed as top-N recommendations.

### 5 Proposed HBSADE model

Since challenges are motivating and pave the way to generating a novel deep learning-based recommender system, which is the combination of both traditional and deep learning methods, our proposed method leverages the resources available, so that the performance of the recommendation increases. Multilayer perceptrons are applied here to train the set of input–output combinations and to learn the dependencies between them. In order to minimize the error value, training involves adjusting weights and bias values. Adjustments to the weights and biases are carried out with proper training methodology. The proposed work comprises a two-stage process. The first is “candidate generation” and the second is “candidate ranking.”

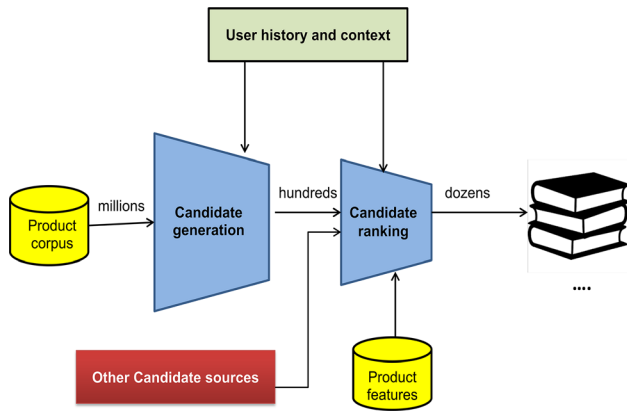


Fig. 9 Two-stage process of the proposed framework

Figure 9 shows an overall idea about the proposal covering the two-stage process.

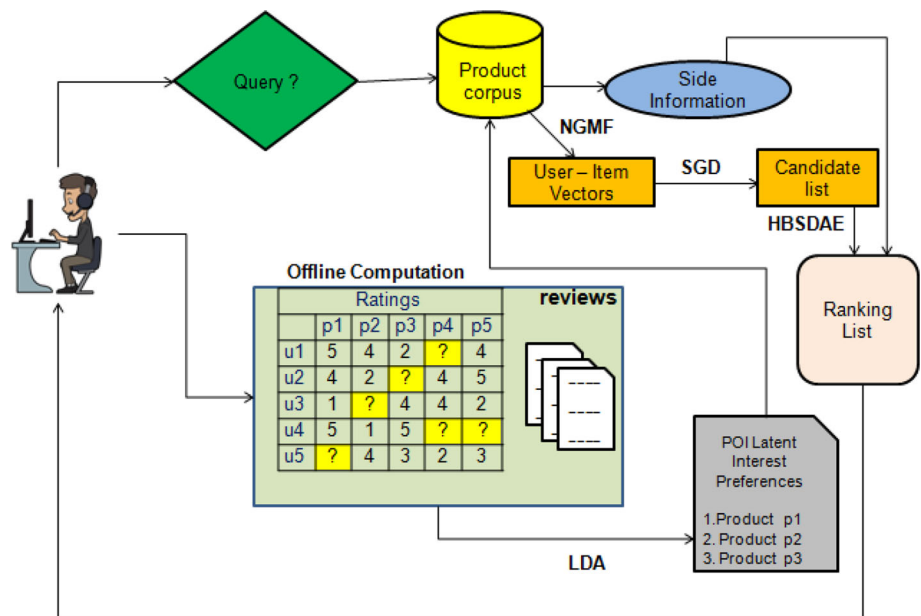
Millions of products’ information is initially available in the product corpus. For example, the symmetric form of embedding layer uses the database of textual reviews assigned to item  $V$ . Concerning the vocabulary highly frequent words which are added and to fix the  $f$ -dimensional vector, the embedding layer will represent the map of each word available in  $\emptyset$ . The refer corpus and the operation of embedding for the contextual document  $\gamma$  are identical.

After analyzing reviews in terms of user/item matrix, the latent interest of the user is generated. During the offline

computation, ratings and reviews are analyzed to convert these into latent interests using the algorithm called LDA. Once the query is generated by the user, “side information” includes the background and demographic details explored from the user. In the very first stage, our proposed method explores the latent interest of the users via a sparse kind of LDA and then extracts the knowledge of low-rank characteristic vector values of users and items through NGMF. Later, using SGD, we will obtain the optimized generation of the candidate list with ranking. When the second-stage activities begin, our proposed method executes candidates ranking process through a BSADE.

To increase the recommendation achievement, the proposed work handles candidate ranking by analyzing side information. HBSADE provides a pair-wise ranking technique used to assess the user–item communications from implicit kinds of feedback. This model/design is the best option for generating recommendations. The rate of learning is altered, and the first-rate achievement is repeated. This work shows the significant interactions between the users and items. The process of candidate generation and ranking for HBSADE is presented in Algorithm 1. From the illustration in Fig. 10, we can observe that our proposed method generally comprises request–reply behavior, which includes an online query and offline computation.

Fig. 10 Proposed HBSADE methodology



**Algorithm 1 Candidate generation and ranking using HBSADE****Input** : Product corpus**Output** : Recommendation list

- 1: Construct a four-layer hybrid stacked denoising autoencoder deep neural network;
- 2: Load product corpus data into the first layer of the network;
- 3: while(true)
  - Apply the LDA method to create POI latent interest preferences;
  - Add these preferences along with side information (demographic particulars) to the input product corpus ;
  - Repeat steps 4 – 8 until we get the best recommendation result for the input query;
 end while
- 4: Apply the combined model of NGMF and MLP on input data;
  - According to Eq.(3), generate user-item vector.
- 5: Perform Candidate generation for the user-item vector by following key steps :
  - Assume  $C_k$  is Candidate itemset of size  $k$  and  $L_k$  : Frequently occurred list of items
  - for ( $k=1$ ;  $L_k \neq \phi$ ;  $k++$ ) do begin
    - $C_{k+1}$ = candidate generated from  $L_k$ ;
    - For each transaction in database do
      - Increment the count of all candidates in  $C_{k+1}$
    - Generate  $L_{k+1}$  candidates in  $C_{k+1}$ .
  - end
  - Update the Generated List.
- 6: Extract the output from the first layer and give the same as input for the next hidden layer;
  - while(iteration>0)
    - update :  $v_j$ , according to Eq.(14);
    - update:  $u_i$ , according to Eq.(15);
    - Generate Rating matrix using the formula  $R_{i,j} \sim N(u_i^T v_j, C_{ij}^{-1})$ ;
    - Apply SGD to generate optimized candidate list with the high predicted score, according to Eq. (20);
  - end while
- 7: Receive the output of step 6 and give the same as input for the next hidden layer.
  - Apply BSADE method to generate ranking list;
- 8: With the help of the ranking list, generate recommendations with a high score based on the query given by the user.

## 6 Experimental analysis

In this section, we evaluate the performance of the proposed hybrid model of the HBSADE approach with two benchmark datasets, namely Amazon-b and Book-Crossing datasets.<sup>2</sup> Both datasets are used for the book recommendation, and we have compared the performance with four state-of-the-art recommendation algorithms.

### 6.1 Datasets

We used three benchmark datasets from different real-world domains. These datasets have composite information of textual reviews, rating values, descriptions and

numerical scores for each category of a book/product/movie/TV. These datasets comprise ratings for user-item pairs with a numerical value ranging from 1 to 5. In total, there are 22,507,155 ratings and 8,898,041 reviews available in the Amazon-b dataset. On the other hand, there are 4,607,047 ratings and 1,697,533 reviews available in the Amazon-m&t dataset. The last dataset, called Book-Crossing dataset, contains 2,78,858 reviews on 2,71,379 books with 11,49,780 ratings. All these datasets lead to a problem in the form of a user-item matrix with a data sparsity of 99.99%. Outlier data and noise data exist in these datasets. Our first aim is to remove noise data and outlier data from these datasets. Using the proposed method, in the very first step called “candidate generation,” we have removed noise and outlier data. Each dataset is split into two forms, with a ratio of 80:20, in

<sup>2</sup> <http://www2.informatik.uni-freiburg.de/~chiegler/BX/>.

which the training dataset provided for implementation has 80% of the observations and the remaining 20% is used for testing purposes.

## 6.2 Evaluation metrics

The main objective of the proposed recommendation system is to generate top- $N$  recommendations to the end users, so we applied Recall@ $N$  and Precision@ $N$  metrics to evaluate our proposed model of HBSADE. To assess the proposed hybrid model, we have arranged the predicted rating values of all products for each user and recommended the top- $N$  recommendations list to each user.

$$\text{Recall@}N = \frac{A \cap B}{B} \quad (21)$$

$$\text{Precision@}N = \frac{A \cap B}{N} \quad (22)$$

where  $A$  is the number of items the user likes in top- $N$  and  $B$  is the list of various items that are adopted by the user. F1 score or F-measure is another metric applied to evaluate the proposed system. F1 score conveys the balance between precision and recall. F1 is calculated through

$$2 * ((\text{precision} * \text{recall}) / (\text{precision} + \text{recall})) \quad (23)$$

*Normalized root mean square error (NRMSE)* Proposed evaluation is executed with a five-cross-fold validation and the NRMSE formula:

$$\text{NRMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left( \frac{\bar{r}_i - r_i}{d_{\max} - d_{\min}} \right)^2} \quad (24)$$

where  $\bar{r}_i$  is the predicted value and  $r_i$  is the real value.

*Normalized discounted cumulative gain (NDCG)* generates the quantity value of the proposed system's original performance based on a graded relevance score. This value ranges from 0.0 to 1.0.

$$\text{DCG} = \sum_{i=1}^N \frac{\bar{r}_i - 1}{\log_2(i + 1)} \quad (25)$$

$$\text{NDCG} = \frac{\text{DCG}}{\text{IDCG}} \quad (26)$$

where ideal discounted cumulative gain (IDCG) is the peak possible discounted cumulative gain (DCG) value, and the actual relevance of a recommended item is predicted using  $\bar{r}_i$ .

## 6.3 Baseline methods and parameter setting

To assess the performance of our proposed approach by comparing it with the benchmark recommendation algorithms, probabilistic matrix factorization is a model

that performs factorization. Factorization happened on the given user–item matrix [32]. It considers the existence of Gaussian observation noise and Gaussian priors to the latent factor model. The convolution matrix factorization (CMF) model is a composite of the CNN [38]. It provides the environment in which contextual information about the representation of vectors has been extracted, and it can be incorporated into MF. This method typically generates accurate recommendation results. Collaborative-type deep learning methods are stratified form of a deep learning model to achieve deep representation learning for the product information and to generate a collaborative filtering-based recommendation system [33]. Collaborative deep learning (CDL) can produce perfect recommendation results. Model of deep learning for top- $N$  recommendation system is designed with the proper utilization of candidate generation and ranking via CNN architecture, which includes vector learning for items, exploring various user's interests [34, 35, 39]. The massive amount of different forms of side information is collected from the user to generate recommendations accordingly.

Our proposed method, called hybrid Bayesian stacked auto-denoising encoder (HBSADE), is the integrated model of combining the features of PMF and SDAE. It is also an ensemble method—combining the collaborative deep learning process with the regular learning process. The low rank generated from a collaborative filtering method is used to provide top- $N$  recommendations to the user. For all compared models, we have completed the training process with available rating information. We have randomly selected 80% data for the training purpose and the remaining 20% data for testing purposes. The performance of each method was measured and monitored via all baseline methods. For our hybrid model, we set the hyperparameters  $\alpha$ ,  $\beta$ , and  $\lambda$  to 0.2, 0.8 and 0.01, respectively. The rate for learning is also given as an input parameter. And we use masking of noise level 0.3 in order to receive the depraved input  $X_0$  from the clean input  $X_C$  from the massive form of inputs. For the proposed deep learning model, the total number of layers is set to 4 in our experimental evaluation and comparison. In addition, the learned latent factors for both the user and the item are set to 64. We use a drop rate of 0.1 to achieve adaptive regularization and to avoid over-fitting.

During the process of exploring users' innate interests, the textual kinds of reviews are combined further as a document for each and every user. During the preprocessing, we first resolve the problem of removing stop words from the massive volume of contextual review documents. Then, we have selected 200 words from the

review document for analyzing the term frequency (TF)/inverse document frequency (IDF) combinations. Here, we set a drop rate of 0.1 to avoid over-fitting. Smaller values for parameters will produce inaccurate recommendation results, whereas the larger amount of selection of parameters will lead to an over-fitting problem, so, the selection of parameters greatly concentrated on the utilization of both input datasets.

**Table 1** Precision for Book-Crossing dataset

Algorithm	$N = 5$	$N = 10$	$N = 15$	$N = 20$	$N = 25$
PMF	0.107	0.098	0.089	0.085	0.078
CDL	0.190	0.175	0.113	0.123	0.112
CMF	0.185	0.147	0.137	0.124	0.117
DLMR-DAE	0.198	0.195	0.186	0.157	0.158
HBSADE	0.283	0.253	0.256	0.237	0.232

**Table 2** Precision for Amazon-b dataset

Algorithm	$N = 5$	$N = 10$	$N = 15$	$N = 20$	$N = 25$
PMF	0.103	0.093	0.082	0.071	0.069
CDL	0.193	0.186	0.124	0.119	0.118
CMF	0.183	0.164	0.132	0.122	0.116
DLMR-DAE	0.295	0.276	0.253	0.234	0.234
HBSADE	0.312	0.297	0.272	0.258	0.258

**Table 3** Recall for Book-Crossing dataset

Algorithm	$N = 5$	$N = 10$	$N = 15$	$N = 20$	$N = 25$
PMF	0.105	0.114	0.123	0.161	0.134
CDL	0.156	0.177	0.195	0.209	0.217
CMF	0.157	0.168	0.164	0.191	0.197
DLMR-DAE	0.305	0.327	0.344	0.354	0.353
HBSADE	0.327	0.349	0.364	0.382	0.371

**Table 4** Recall for Amazon-b dataset

Algorithm	$N = 5$	$N = 10$	$N = 15$	$N = 20$	$N = 25$
PMF	0.109	0.124	0.138	0.151	0.152
CDL	0.193	0.209	0.224	0.236	0.238
CMF	0.203	0.211	0.223	0.226	0.226
DLMR-DAE	0.318	0.352	0.376	0.381	0.382
HBSADE	0.341	0.376	0.392	0.398	0.399

## 6.4 Comparison and performance evaluation

In this section, we list all the experiments carried out with our proposed approach along with benchmark recommendation methods. Tables 1, 2, 3 and 4 provide a detailed report on comparing the performance of the rating prediction for CDL, PMF, CMF, DLMR-DAE and HBSADE concerning Recall@N and Precision@N. The result of the tables indicate that:

- Using the method called PMF, taking into consideration user–item rating matrix numerical values—Precision@N and Recall@N values are lesser than those other benchmark methods, namely CDL, CMF and DLMR-DAE.
- CDL seeks to improve the performance of the recommendation system by introducing a stacked denoising auto-encoder. Experiment results show that CDL is lightly better than the previous model PMF.
- CMF provides a composite kind of CNN of deep learning to make the environment learn about model vectors for the contextual information, and it can be integrated into MF. This method generates less accurate recommendation results than the previous techniques of PMF and CDL.
- The results generated from the DLMR-DAE are slightly better than the results obtained from PMF, CDL and CMF. As stated earlier, DLMR-DAE works to explore the innate interests of the user.
- Our proposed approach HBSADE outperforms the existing benchmark methods of PMF, CDL, CMF and DLMR-DAE. Using the approach, HBSADE, the learned interests and textual descriptions, such as reviews, are applied to candidate generation and candidate ranking. By the end of these two phases, we have achieved better results. User’s side information and top-N recommended list of items mainly focused on producing better results. The arrived precision value and recall value are quite better than the existing methods.
- By comparing the results, we got the inferences, saying that precision values for Amazon-b are quite better than for the Book-Crossing dataset.

The experiments of the proposed system HBSADE were conducted on two large-scale real-world datasets and attained results compared with the traditional recommendation techniques for the evaluation purpose. The comparison and analysis of the experimental results obtained by PMF, CMF, CDL and DLMR-DAE are given in Fig. 11 which shows the results obtained for the Book-Crossing dataset using the precision metric. Figure 12 depicts the attained precision results of the proposed work for the

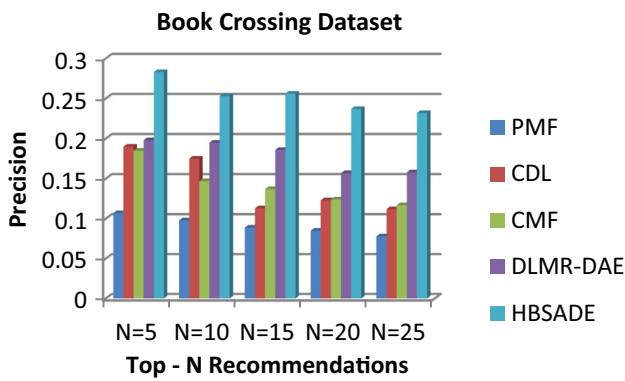


Fig. 11 Precision value comparison of hybrid Bayesian stacked denoising auto-encoder with other recommender systems for Book-Crossing dataset

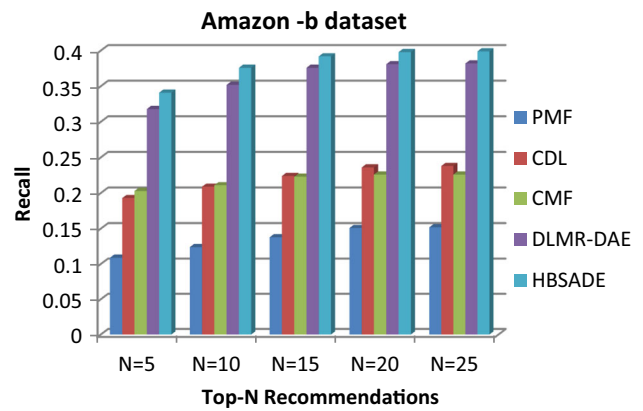


Fig. 14 Recall value comparison of hybrid Bayesian stacked denoising auto-encoder with other recommender systems for Amazon-b dataset

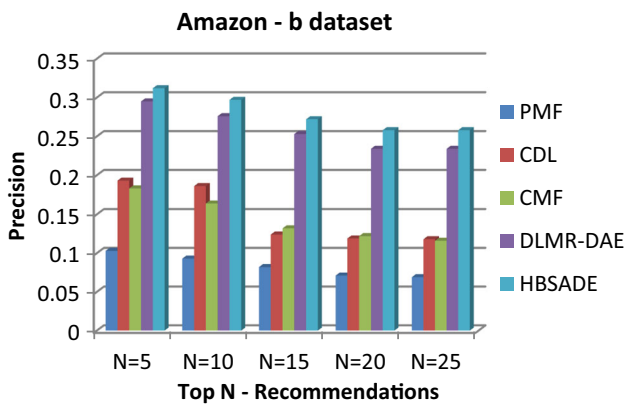


Fig. 12 Precision value comparison of hybrid Bayesian stacked denoising auto-encoder with other recommender systems for Amazon-b dataset

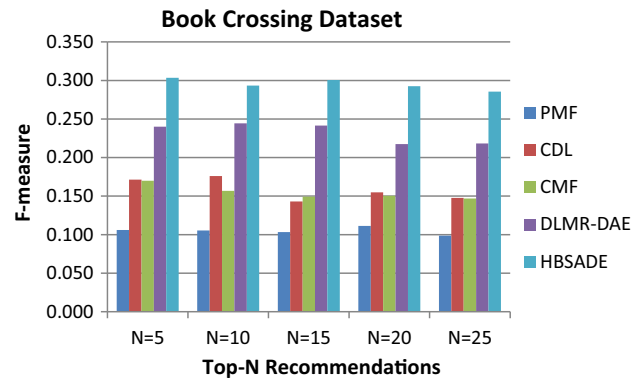


Fig. 15 F-measure value comparison of hybrid Bayesian stacked denoising auto-encoder with other recommender systems for Book-Crossing dataset

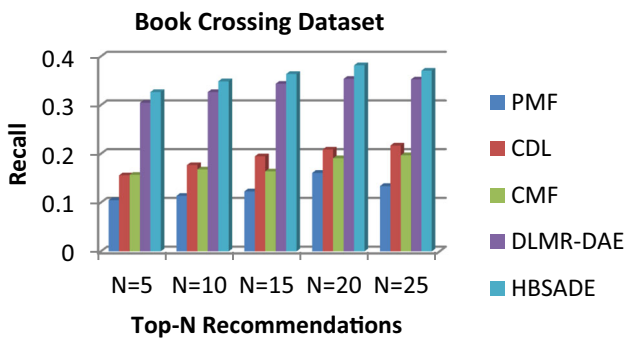


Fig. 13 Recall value comparison of hybrid Bayesian stacked denoising auto-encoder with other recommender systems for Book-Crossing dataset

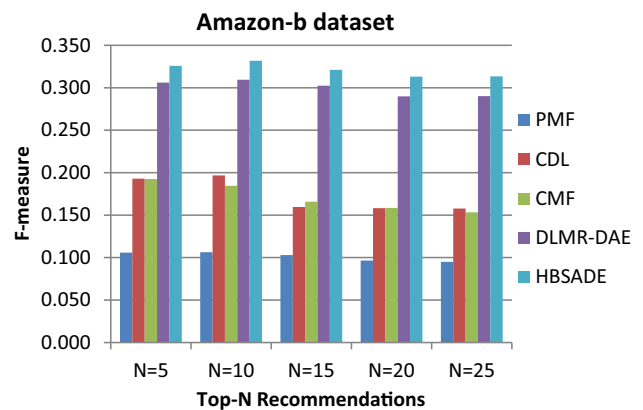
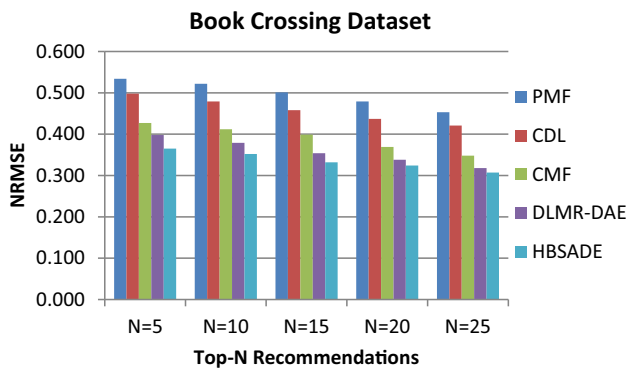


Fig. 16 F-measure value comparison of hybrid Bayesian stacked denoising auto-encoder with other recommender systems for Amazon-b dataset

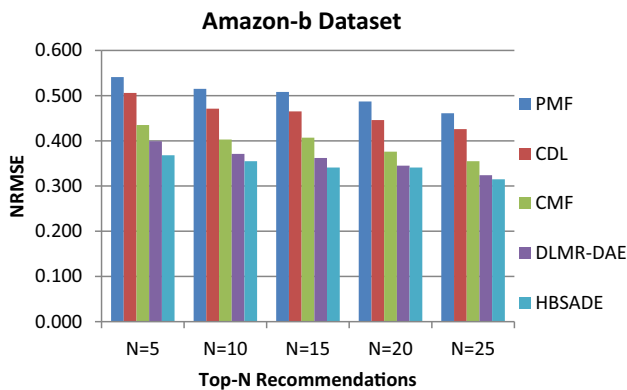
Amazon-b dataset compared with other existing approaches.

Achieved values of Precision@N decrease slowly with increasing N value. Figure 13 shows recall metric results

obtained using the Book-Crossing dataset, and Fig. 14 shows the achieved results of the proposed work for the



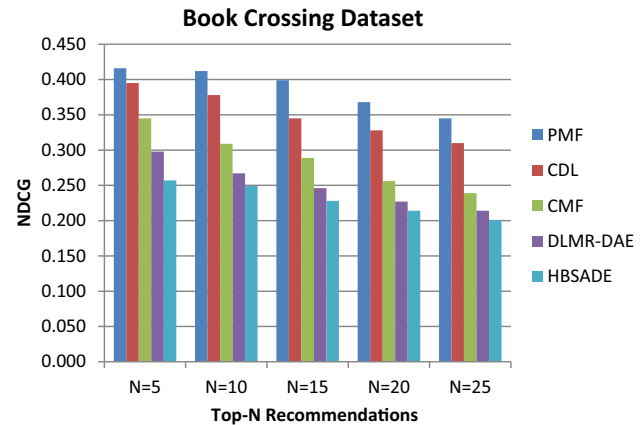
**Fig. 17** NRMSE value comparison of hybrid Bayesian stacked denoising auto-encoder with other recommender systems for Book-Crossing dataset



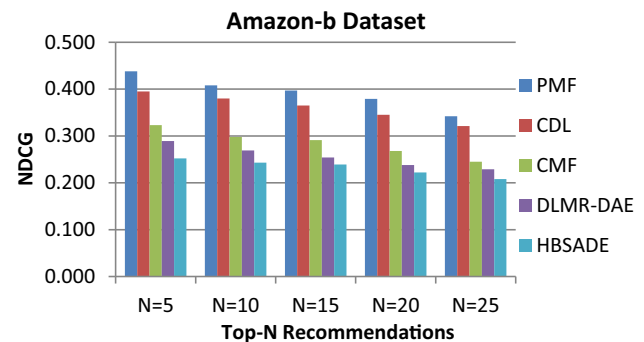
**Fig. 18** NRMSE value comparison of hybrid Bayesian stacked denoising auto-encoder with other recommender systems for Amazon-b dataset

Amazon-b dataset. The value of Recall@N increases gradually along with increasing N. Attained results of Recall@N and Precision@N for the method called PMF are relatively small when compared with CTR, CDL, CMF, DLMR-DAE and HBSADE.

Figure 15 shows the results obtained for the Book-Crossing dataset using the f-measure metric, and Fig. 16 depicts the attained f-measure results of proposed work for the Amazon-b dataset compared with other existing approaches. The value f-measure decreases gradually with the increasing N recommendations. Achieved results of F-measure@N with Amazon-b dataset have more or less the same set of values than with the Book-Crossing dataset. Values of Precision@N, Recall@N and F-measure@N share a similar drift jointly for these techniques over each dataset. HBSADE outperforms PMF, CDL, CMF and DLMR-DAE naturally in terms of Precision@N, Recall@N and F-measure@N over Amazon-b and Book-Crossing datasets. CDL and CMF work to enhance the



**Fig. 19** NDCG value comparison of hybrid Bayesian stacked denoising auto-encoder with other recommender systems for Book-Crossing dataset



**Fig. 20** NDCG value comparison of hybrid Bayesian stacked denoising auto-encoder with other recommender systems for Amazon-b dataset

performance of recommendations through adding topic regression module, and CDL and CMF perform slightly better than PMF.

A comparison of the evaluation metric NRMSE using the Book-Crossing dataset for PMF, CMF, CDL, DLMR-DAE and HBSADE is reported in Fig. 17. Similarly, a comparison of the NRMSE evaluation metric for the Amazon-b dataset is depicted in Fig. 18. The values of NRMSE over Amazon-b are slightly larger than that of the Book-Crossing dataset.

A comparison of evaluation metric NDCG using the Book-Crossing dataset for PMF, CMF, CDL, DLMR-DAE, and HBSADE is shown in Fig. 19. The contrast of the NDCG evaluation metric for the Amazon-b dataset is shown in Fig. 20. The achieved results of NRMSE and NDCG for PMF among each dataset are much more significant than those of other methods, respectively, since PMF considers only the numerical form of a user-item matrix and discards additional available information.

Overall, each recommendation approach, along with top-N recommendation with  $N = 10$ , is lightly better than



with  $N = 5$ . Similarly, top- $N$  advice with  $N = 20$  is lightly better than with  $N = 10$ . It means that the proposed model is capable of providing recommendations than traditional recommendation approaches. We carried out the list of experiments between two datasets to compare the effectiveness of HBSADE with conventional approaches, including Precision@ $N$ , Recall@ $N$ , F-measure@ $N$ , NRMSE and NDCG. Based on the existing research, the proposed method HBSADE outperforms PMF, CDL and CMF over Amazon-b and Book-Crossing datasets. The obtained results depict the improved performance of the proposed HBSADE model over the traditional recommendation methods. The proposed method HBSADE seeks to extract the latent interests for each user, and then CMF has been performed for candidate generation which includes latent interests and textual information.

In summary, from the analysis of Amazon-b and Book-Crossing dataset, we could understand that the enhanced performance of HBSADE is stable and effective over real-world datasets. It can generate efficient and accurate top- $N$  recommendations in contrast to the traditional recommendation systems. All experiments were conducted in the programming model “Python” on a Personal Computer with Intel i7-8700K supported CPU and NVIDIA graphics card supported GTX1080Ti GPU-based system. For Amazon-b and Book-Crossing datasets, it requires 200 epochs for HBSADE to achieve real convergence in the training phase.

## 7 Conclusion and future work

In RS, data sparsity is an open and challenging issue. Existing methodologies were failed to handle the sparsity problem due to the generation of noise data and the form of outliers in the side information. Initially, the side information mitigates the issue of data sparsity. In this article, we proposed a novel deep learning model called HBSADE, which has been used to eliminate the data sparsity and the removal of outliers, such as noise data. Explicit rating, implicit rating and side information are integrated to learn the latent interest of the user. To capture the explicit rating information, we have applied the HBSADE model that explores the distribution of user’s interests via CNN and performs convolution matrix factorization along with an optimization algorithm SGD. The proposed model has been applied to learn low-rank feature vectors for both users and items. Next, the prediction has been attained for candidate generation. A three-layer hybrid stack-based denoising auto-encoder with heterogeneous size information was applied to handle the problem of data sparsity. Using the approach, HBSADE, the learned interests and textual descriptions, such as reviews, are applied to candidate

generation and candidate ranking. HBSADE outperforms the existing benchmark methods of PMF, CDL, CMF and DLMR-DAE. We have evaluated our model with various evaluation metrics—Precision@ $N$ , Recall@ $N$ , F-measure@ $N$ , NRMSE and NDCG. The performance analysis shows that top- $N$  recommendations obtained from HBSADE outperform other traditional methods in terms of real-world datasets, namely Amazon-b and Book-Crossing. In the future, we are planning to add time-sequence information and behavioral information with the help of a social media network that enhances the module of interest exploring and improves the performance of RS.

**Acknowledgements** The authors gratefully acknowledge the Science and Engineering Research Board (SERB), Department of Science and Technology, India, for the financial support through Mathematical Research Impact Centric Support (MATRICS) scheme (MTR/2019/000542). The authors also acknowledge SASTRA Deemed University, Thanjavur, for extending infrastructural support to carry out this research work.

## Compliance with ethical standards

**Conflict of interest** The authors declare that there is no conflict of interest regarding the publication of this article.

## References

- Hu QY, Zhao ZL, Wang CD, Lai JH (2017) An item orientated recommendation algorithm from the multi-view perspective. *Neurocomputing* 269:261–272
- Hu QY, Huang L, Wang CD, Chao HY (2019) Item orientated recommendation by multi-view intact space learning with overlapping. *Knowl-Based Syst* 164:358–370
- Zhang W, Zou H, Luo L, Liu Q, Wu W, Xiao W (2016) Predicting potential side effects of drugs by recommender methods and ensemble learning. *Neurocomputing* 173:979–987
- Logesh R, Subramaniaswamy V, Malathi D, Sivaramakrishnan N, Vijayakumar V (2019) Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-018-3891-5>
- Balabanovic M, Shoham Y (1997) Content-based, collaborative recommendation. *Commun ACM* 40(3):66–72
- Sarwar BM, Karypis G, Konstan JA, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th international conference on World Wide Web*, pp 285–295
- Zhao ZL, Wang CD, Lai JH (2016) AUI&GIV: recommendation with asymmetric user influence and global importance value. *PLoS ONE* 11(2):e0147944
- Xue HJ, Dai X, Zhang J, Huang S, Chen J (2017) Deep matrix factorization models for recommender systems. In: *IJCAI*, pp 3203–3209
- Burke RD (2002) Hybrid recommender systems: survey and experiments. *User Model User-Adap Inter* 12(4):331–370
- Salakhutdinov R, Mnih A (2007) Probabilistic matrix factorization. In: *NIPS*, pp 1257–1264
- Wang H, Wang N, Yeung DY (2015) Collaborative deep learning for recommender systems. In: *KDD*, pp 1235–1244

12. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res* 11:3371–3408
13. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37. <https://doi.org/10.1109/MC.2009.263>
14. Andreas M (2017) Matrix factorization techniques for recommender systems. Ph.D. thesis, The University of Aegean
15. Salakhutdinov R, Mnih A, Hinton G (2007) Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the 24th international conference on machine learning, pp 791–798
16. Gao J, Pantel P, Gamon M, He X, Deng L (2014) Modeling interestingness with deep neural networks. In: Proceedings of the conference on empirical methods natural language process, pp 2–13
17. Wu Y, DuBois C, Zheng AX, Ester M (2016) Collaborative denoising auto-encoders for top-n recommender systems. In: Proceedings of the ninth ACM international conference on web search and data mining, pp 153–162
18. Kim D, Park C, Oh J, Lee S, Yu H (2016) Convolutional matrix factorization for document context-aware recommendation. In: Proceedings of the 10th ACM conference on recommender systems, pp 233–240
19. Zhang S, Yao L, Sun A, Tay Y (2019) Deep learning based recommender system: a survey and new perspectives. *ACM Comput Surv (CSUR)* 52(1):1–38
20. Miiikkulainen R, Liang J, Meyerson E, Rawal A, Fink D, Francon O, Raju B, Shahrzad H, Navruzyan A, Duffy N, Hodjat B (2019) Evolving deep neural networks. *Artificial intelligence in the age of neural networks and brain computing*. Academic Press, London, pp 293–312
21. Van den Oord A, Dieleman S, Schrauwen B (2013) Deep content-based music recommendation. In: *Advances in neural information processing systems*, pp 2643–2651
22. Bebis G, Michael G (1994) Feed-forward neural networks. *IEEE Potentials* 13(4):27–31
23. Zhang W, Du Y, Yoshida T, Yang Y (2019) DeepRec: a deep neural network approach to recommendation with item embedding and weighted loss function. *Inf Sci* 470:121–140
24. He X, Chua TS (2017) Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, pp 355–364
25. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2012) BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the conference on uncertainty in artificial intelligence, pp 452–461
26. Krestel R, Fankhauser P, Nejdl W (2009) Latent Dirichlet allocation for tag recommendation. In: Proceedings of the third ACM conference on recommender systems, pp 61–68
27. Ren X, Song M, Haihong E, Song J (2017) Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation. *Neurocomputing* 241(7):38–55
28. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web, pp 173–182
29. Sedhain S, Menon AK, Sanner S, Xie L (2015) Autorec: autoencoders meet collaborative filtering. In: Proceedings of the 24th international conference on World Wide Web, pp 111–112
30. Wu X, Yuan X, Duan C, Wu J (2019) A novel collaborative filtering algorithm of machine learning by integrating restricted Boltzmann machine and trust information. *Neural Comput Appl* 31(9):4685–4692
31. Li S, Kawale J, Fu Y (2015) Deep collaborative filtering via marginalized denoising auto-encoder. In: Proceedings of the 24th ACM international on conference on information and knowledge management, pp 811–820
32. Salakhutdinov R, Mnih A (2008) Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In: Proceedings of the 25th international conference on machine learning, pp 880–887
33. Wang H, Wang N, Yeung DY (2015) Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1235–1244
34. Zhou W, Li J, Zhang M, Wang Y, Shah F (2018) Deep learning modeling for top-N recommendation with interests exploring. *IEEE Access* 6:51440–51455
35. Vilorio A, Li J, Guilianny JG, de la Hoz B (2020) Predictive model for detecting customer's purchasing behavior using data mining. In: Proceedings of 6th international conference on big data and cloud computing challenges, pp 45–54
36. Maind SB, Wankar P (2014) Research paper on basic of artificial neural network. *Int J Recent Innov rends Comput Commun* 2(1):96–100
37. Abedini F, Menhaj MB, Keyvanpour MR (2019) An MLP-based representation of neural tensor networks for the RDF data models. *Neural Comput Appl* 31(2):1135–1144
38. Hemanth DJ, Deperlioglu O, Kose U (2020) An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network. *Neural Comput Appl* 32(3):707–721
39. Bell R, Volinsky C (2010) Matrix factorization for recommender systems. Presentation at UMBC

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.