Computers, Materials & Continua
DOI: 10.32604/cmc.2023.032856
Article

Improved Hybrid Deep Collaborative Filtering Approach for True Recommendations

Muhammad Ibrahim[1], Imran Sarwar Bajwa[1], Nadeem Sarwar[2,*], Haroon Abdul Waheed[3], Muhammad Zulkifl Hasan[4] and Muhammad Zunnurain Hussain[4]

[1]Department of Computer Science, The Islamia University of Bahawalpur, Bahawalpur, Pakistan
[2]Department of Computer Science, Bahria University Lahore Campus, Lahore, Pakistan
[3]Software Engineering Department, Faculty of IT, University of Central Punjab, Lahore, Pakistan
[4]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Selangor, Malaysia
*Corresponding Author: Nadeem Sarwar. Email: Nadeem_srwr@yahoo.com
Received: 31 May 2022; Accepted: 17 September 2022

Abstract: Recommendation services become an essential and hot research topic for researchers nowadays. Social data such as Reviews play an important role in the recommendation of the products. Improvement was achieved by deep learning approaches for capturing user and product information from a short text. However, such previously used approaches do not fairly and efficiently incorporate users' preferences and product characteristics. The proposed novel Hybrid Deep Collaborative Filtering (HDCF) model combines deep learning capabilities and deep interaction modeling with high performance for True Recommendations. To overcome the cold start problem, the new overall rating is generated by aggregating the Deep Multivariate Rating DMR (Votes, Likes, Stars, and Sentiment scores of reviews) from different external data sources because different sites have different rating scores about the same product that make confusion for the user to make a decision, either product is truly popular or not. The proposed novel HDCF model consists of four major modules such as User Product Attention, Deep Collaborative Filtering, Neural Sentiment Classifier, and Deep Multivariate Rating (UPA-DCF + NSC + DMR) to solve the addressed problems. Experimental results demonstrate that our novel model is outperforming state-of-the-art IMDb, Yelp2013, and Yelp2014 datasets for the true top-n recommendation of products using HDCF to increase the accuracy, confidence, and trust of recommendation services.

Keywords: Neural sentiment classification; user product attention; deep collaborative filtering; multivariate rating; artificial intelligence

1 Introduction

Recommendation systems (RSs) are software agents that automatically suggest products according to user interests or preferences. Recommendation system services are based on *Arterial*

Intelligence (AI), *Natural Language Processing* (NLP), and *Machine Leering* (ML) or *Deep Learning* (DL) which provide empirical solutions and got great importance in facilitating the decision-making process across the various real-world applications such as e-commerce [1], e-tourism [2], e-healthcare [3], e-purchasing [4], e-game [5] and so many others. Users may express their opinions in the form of ratings, votes, likes, and reviews about products on social media. Several user behaviors such as searching, spending time, watching [6], and the historical sequence of interaction with products are *Implicit Feedback* while likes, votes, and stars to a product or features of products are *Explicit Feedback* [7] and [8]. So, the rating that is explicitly given by users is more reliable than implicitly. That is the reason we use explicit feedback. Some of the recommenders using *Uni-Variant* (IMDB, Rotten Tomatoes, Netflix) while another's using *Bi-Variant* (Facebook and Fandango) rating is not suitable and significant for measuring the popularity scores of products for recommendations because different sites have different ratings about the same product[1]. Therefore, there is a need to evaluate the true weight of product popularity for top-n ranking in recommendations by using the multivariate (likes, votes, stars and sentiment scores) from different sites [9] and [10].

Usually, a recommendation system is based on *Content-based Filtering* (CB), *Collaborative based Filtering* (CF), or *hybrid-based Filtering* (HF) [11] and [12]. CF techniques have been extensively researched for personalized recommendations based on the interaction between users and products instead of using the previous history or knowledge about users and products. CF techniques are simple and efficient but suffer from cold start problems, the accuracy of prediction, and do not have the maturity to capture the complex relationship (Interaction) between user and product. Most of the conventional CF techniques are based on the *Factorization of Matrix* (MF) [12] in which users and products are characterized by *Latent Factors* (LF) derived from the rating matrix for user products. Traditionally, in *Latent Factor Matrix (*LFM) models in CF [13], the user's choices for a product are frequently predicted with a linear kernel, like a *Dot Product* of its Latent Factors, but the complex structure of user-product interaction is not handled effectively. Currently, Deep Learning based recommendation systems [14] and [15] overcome the limitations of traditional recommendation system approaches, such as complex user preferences, product features, and their relationships themselves to gain high performance in the recommendation. Most auxiliary information is explored by deep learning for recommendations, such as modeling the product features [16] and [17]. Recent research is mainly continuing to use conventional MF-based methods for the user-product rating matrix. Firstly, *Restricted Boltz Mann Machines* (RBMs) [18] using the two-layer neural network instead of deep learning for user-product rating matrix is modeled to achieve more accurate results over conventional approaches. Another recent method, *Collaborative Denoising Auto-Encoder* (CDAE) [19] is seemed to primarily design the rating prediction model by implementing the one hidden layer between the input and output layer. While *Neural Collaborative Filtering* (NCF) [20] is designed the interaction model by employing the multi-layer perceptrons, However, this does not analyze any preferences of users and characteristics of products which appear to be useful in maximizing CF performance in the recommendation. Rather than explicit feedback on results, NCF and CDAE use only implicit feedback for recommendations. *Deep Matrix Factorization* (DMF) [21] is designed for the user-product rating matrix in which features of users and products are mapped into a low-dimensional with nonlinear representations by using the deep neural networks in which user-product interactions are computed by an inner product and as the LFM [13], apply similarly as the linear kernel (i.e., Dot function). It is assumed that the effectiveness can be gained by capturing the users and product information as well as both non-linear and non-trivial user-product relationship with multi-layer projection [22].

[1] https://www.freecodecamp.org/news/whose-reviews-should-you-trust-IMDb-rotten-tomatoes-Metacritic-or-fandango-7d1010c6cf19/

A recommendation system based on *Convolutional Neural Networks* (CNN) is used to extract the product features from short text or context information using ontologies as axillary information [16], On the other hand, the recommender system based on "*Recurrent Neural Networks*" (RNNs) is used to explore the sequential features and temporal dynamics of ratings [23]. But they only focus on the content of text typically while users and products have crucial influences on sentiment classification. For example, different users may express their different emotional intensity with the same words; lenient users use "good" words for ordinary products while potential users may use this word for excellent behavior about the product. Likewise, review ratings may also affect a product's characteristics. Higher quality products lead to higher ratings while low-quality product leads to low rating [24]. Enables the Sentiment Classification by incorporating user and product details at the word level to reflect preferential matrix and vector distribution for each user and product of CNN Sentiment Classification. It has some improvements but it represents the information of users and product at the Word Level instead of the Document Level.

User and Product information are incorporated to overcome the issues [25] by using the attention mechanism in a better way. User and product embedding are used to present the user's preferences and product characteristics respectively are achieved by using the joint learning and User Product Attention (UPA) mechanism hierarchically [25]. But it does not implement the user-product interaction model for collaborative filtering in recommendations. *Joint Neural Collaborative Filtering* (J-NCF) [26] proposed the collaborative model that is based on the extraction of features and user-product interaction modules using the MLP that would be jointly trained to form a unified deep learning structure consisting of two neural networks for the recommendation. But this model (J-NCF) does not use the *Hierarchical Attention Network* (HAN) [27] mechanism by which user's preferences and product characteristics are incorporated for document representation that is further used for interaction modeling between user and product for more accurate prediction.

As we previously addressed, the major shortcoming and issues in different recommendation systems, the true popularity of the product, users and product documentation representation, and sentiment class extraction from reviews by using the incorporation of user's preferences and product characteristics are the shortcoming of J-NCF [26], while NCF [20] is a shortcoming of [25]. So, our novel proposed model HDCF provides the empirical solution to overcome the limitations of [25], by integrating UPA, DCF, NSC, and DMR. The proposed model will learn the user's presence and product characteristics from reviews and encode the reviews into document level representation using the attention of user and product, further this representation fed as input to interaction model and sentiment classifier. Sentiment classifier predicts the favorability class of reviews. On the other hand, the overall rating is improved by fusing the discrete rating (Likes, Votes, and Stars) with continuous values (Sentiment) of reviews to make the DMR [9] for improving the accuracy of the popularity of the product in the recommendation system. User and product information for document representation will be frequently used for training and classification which is the reason we combined the strategy our recommendations are the following:

- Using the continuous (short text or reviews) and discrete rating (votes, likes, and stars) using deep learning for multivariant rating.
- Ranking the product based on multivariant rating instead of uni-variant or bi-variant
- Scrape Rating from multi-data sources or sites because different sites have different ratings for the same product which makes confusion or ambiguity for product ranking.
- Generate the user preferences using the user-product attention mechanism.
- Generate the overall rating using Multivariant Ing instead of multicriteria rating for cold star problem.
- Integrate the NFC + NSC + UPA with DMR to propose the improved HDCF.

So, we can say that our contribution to the state-of-the-art recommendation systems is by better utilization of UPA, DCF, NSC, and DMR for the user to make better product suggestions.

2 Related Work

Typically, most of the research is based on the factorization of the user-product rating matrix [13] with *Singular Value Decomposition* (SVD) using the *Inner Product* between both low-rank matrices, one for user factors and another one for product factors. The user's preferences are generated as follows: $\hat{y}_{up} = \left(\mathrm{b}^{up} + (\mathrm{d}^{u}.\mathrm{d}^{p})^{T}\right)$ here d^u and d^p denote the inner product of factors user and product respectively while b^{up} the bias. Traditional SVD is unable to define the unknown (missing) rating. Most of the researcher tries to overcome the problem by baseline estimation [28] but Which results in a dense matrix of user ratings and complex factorization in terms of infeasible computation. Recently objective functions are used to minimize the prediction error to learn factor vectors directly on unknown ratings for avoiding overfitting. Usually, SVD approaches are used for new users after giving their rating to particular products without rebuilding the parameters in the models to minimize the objective function by using gradient descent. So, SVD using the current rating provides recommendations for new users. *Matrix Factorization* (MF) model performance is improved by incorporating Explicit and implicit feedback SVD++ [13]. That is the reason for sparse rating matrices by using the typical MF methods with complex computational costs for the decomposition of the rating matrix. Mostly conventional RSs are implemented through a *Linear kernel* to design the interactions between user and product using an *Inner Product* vector of users and products. Linear functions may not have been able to provide significant information on the user characteristics (products) and user-product interactions: earlier work has shown that nonlinearities have potential benefits with systematic studies in maximizing the performance of RSs [29], [30], and [19]. RSs that are based on *Deep Learning* (DL) are classified into two brought categories as *Single Neural networks* (SNN) and integrational models. In a single neural network, RBM [18] is the earliest neural recommender system that models the tabular data explicit rating of products by using the two-layer undirected graph. RBM focuses on the prediction of ratings not on top-n recommendations while its loss function is used only for known ratings [19]. Rating prediction is computed by *Auto-Encoder* (AutoRec) [30] which uses the loss function only for known ratings, which means it is not good for providing the accuracy for top-n recommendations. Auto-encoder fails for generalization of unknown rating data, auto-encoder is prevented from learning the identity function by using the *Denoising Autoencoders* (DAE) [29] to learn from corrupted inputs intentionally. CDAE [29] uses implicit feedback that is partially observed as inputs. Unlike our model, DAEs and CDAEs use a product-product model for personalizing recommendation is represented by ratings of products given by a user, and the product values are decoded by learning the user's representation for outputs. Unlike previous models, our model is a type of user-product model in which users and product representation are learned first then measures the correlation.

The state-of-the-art NCF implements the multi-layer perceptron to design the uses-products interaction model that represents the non-linear relationship between users and products [20]. However, user and product representation are initialized in a limited manner by using the one-hot vector for users and products. While j-NCF [26] uses the two neural networks DF and DI for users and product representation respectively that are further concatenated for the input of the interaction model. While our proposed model uses the same sense in a better way because "*User Product Attention*" (UPA) in a hierarchical manner captures the users and product features and their relationship more accurately than NCF and JNCF. *Cross-Domain Content Boosted Collaborative Neural Networks* (CCCFNet) [31] based on the dual network one for users and another for products using the content information to explore the user's preferences and product features, further to compute the user-product interaction with dot product in the last layer. Faultlessly in DeepFM [32], The integration of the factorization machine and multi-layer perceptron is modeled as end-to-end and uses the content information for low-order interactions using a factorization machine while using deep neural networks for higher-order feature interaction. Unlike DeepFM, our proposed model adopts the rating information from reviews or short text to explore reliably both information of users and product by using the hierarchical attention mechanism. *Collaborative Deep Learning* (CDL) [17] proposed the deep integration model based on the *Hierarchical Bayesian Model* (HBM) in which DAEs stack is integrated into conventional probabilistic MF. It has two deviations from our proposed model; the first one is the deep features representations of the product are extracted from content information and another one is to model the user-product relationships via the *Dot Product* of vectors of user and product employing linear kernel. Another popular model for integration is *Deep Cooperative Neural Networks* (DeepCoNNs) [22] which uses *Convolutional Neural Networks* (CNNs) to extract the behavior of users and characteristics of the product from reviews. It applies the factorization machine for interactions between users and products from predictions of rating to overcome the problem of sparsity. Another integration model, *Deep Matrix Factorization* (DMF) [21] is based on a DNN that transforms the users and product feature matrix into a space of low dimensional that follows the LFM which computes the interaction between users and product by using the *Inner Product* of user and product. HBSADE [33] is a framework that has been used to remove data sparsity and outliers, such as noise data. To learn the user's latent interest, explicit rating, implicit rating, and side information are combined. We used the HBSADE model, which uses CNN to examine the distribution of user interests and performs convolution matrix factorization combined with the optimization algorithm SGD to capture explicit rating information. The TADCF [34] recommendation model consists of two CF recommendation mechanisms. A time-aware attention mechanism is used in the 1st stage to model dynamic user preferences, in which all the predicted products, recently interacted products, and their interaction time is all taken into account to determine the importance of different historically interacted products for short-term preference modeling. The dynamic user preference model is then built by combining short- and long-term user preferences. The linear and nonlinear user-item feature interactions are learned in the second stage using a matching function learning model based on two types of DL models, DMF and DNN. Finally, the feature interaction vectors of the two models are fused in the TADCF model's final layer to generate the user-product matching score. Unlike DMF, we adopt the multilayer perceptrons to model the user-product interaction using the concatenation of user and product features as input that is extracted by using the UPA mechanism by incorporating the user and product information to improve the accuracy more expressively. As we addressed the previous work limitations, our novel proposed work is the empirical solution for that problem. Our proposed model jointly learns the features and the design of the interaction model into an *End to End* trainable neural network. The [20] NCF that don't user-product attention mechanism while [26] J-NFC uses the User Product Attention jointly [9]. TFIDF + MR uses the TFIDF weighting scheme for sentiment scoring as well as just using the multivariant rating and doesn't use the NSC and NFC [22]. DeepCoNN uses the deep convolutional neural network instead of RNN/LSTM variations as well as did not use the multivariant rating and user product attention for user preferences learning [10]. NSC + MR combined the Neural sentiment classifier with a Multivariant rating to achieve some improvements. HDCF combined the NCF + UPA + NSC + DMR gained more improvement as compared to baseline models.

Here the following Table 1 represents the comparisons and research gaps in the recommendation's models as addressed previously.

Table 1: Research gaps matrix of recommendation models

References	HAN	DCF	NSC	Discrete rating	Continuous rating	Multivariate rating
[21] DMF	×	✓	×	✓	×	×
[20] NCF	×	✓	×	✓	×	×
[26] J-NFC	×	✓	×	✓	×	×
[25] NSC + UPA	✓	×	×	×	✓	×
[9] TFIDF + MR	×	×	×	✓	✓	✓
[34] TADCF	✓	✓	×	×	✓	×
[10] NSC + MR	✓	✓	×	✓	×	×
HDCF	✓	✓	✓	✓	✓	✓

3 Design Model

The proposed novel model is based on four major components; first, user and product document embedding by using UPA, second, NSC of reviews about the product, third, the novel Multivariant Rating (stars, votes, likes, and sentiment) about a product is obtained by combining the continues rating (sentiment of reviews) and discrete rating (stars, votes, and likes) and fourth, DCF is used for User-Product Interaction by using DMR, MLP. UPA is used for encoding the information of the user and product by incorporating the preferences of the user and the characteristics of the product. UPA uses BiLSTM for word and sentence level encoding of the reviews to get explicit document representation. Further, this document representation is fed into the interaction module to extract the interaction between users and products. Document representation is also fed into the NSC module in which sentiment classification is performed. The NSC is defined as achieving the User Product Attention, The sentiment classifier transforms the reviews (continuous sentiment) into the real number (discrete sentiment) and classifies the discrete sentiment into five classes by labeling the favorability status. DMR is computed by combining the Votes, Stars, Likes, and Sentiment scores to generate the popularity status of product labeling with the medal for user convenience to easily identify the most popular product. The following Fig. 1. Illustrates the architecture of the novel proposed HDCF.

3.1 Document Representation

Here the formal representation of the document is represented by these nomenclatures used such as a user $u \in U$ writing a review about a product $p \in P$ while the review is denoted by $d \in \mathrm{D}$ with n sentences $d = (\mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_n)$ where l_i is the length of i^{th} sentence, with l_i words $\mathrm{S}_i = \left(w_1^i, w_2^i, \ldots, w_{l_i}^i\right)$. We employ a hierarchal structure with the bidirectional LSTM to obtain the document representation for modeling the user-product interaction and document-level text semantics. Word embedding $\mathrm{w}_j^i \in \mathbb{R}^d$ are employed to each word w_j^i at the word level. $\left(\mathrm{w}_1^i, \mathrm{w}_2^i, \ldots, \mathrm{w}_{l_i}^i\right)$ are received by BiLSTM and hidden states $(\mathrm{h}_1^i, \mathrm{h}_2^i, \ldots, \mathrm{h}_{l_i}^i)$ are generated or fetch the last hidden state h_j^i or sentence representation s_i is obtained by usage of the *Attention* mechanism. Generated sentence representation $(\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_\mathrm{n})$ is fed into BiLSTM hidden states $(\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_n)$ and in a similar way document d^{up} representation is also obtained in the following Fig. 2.

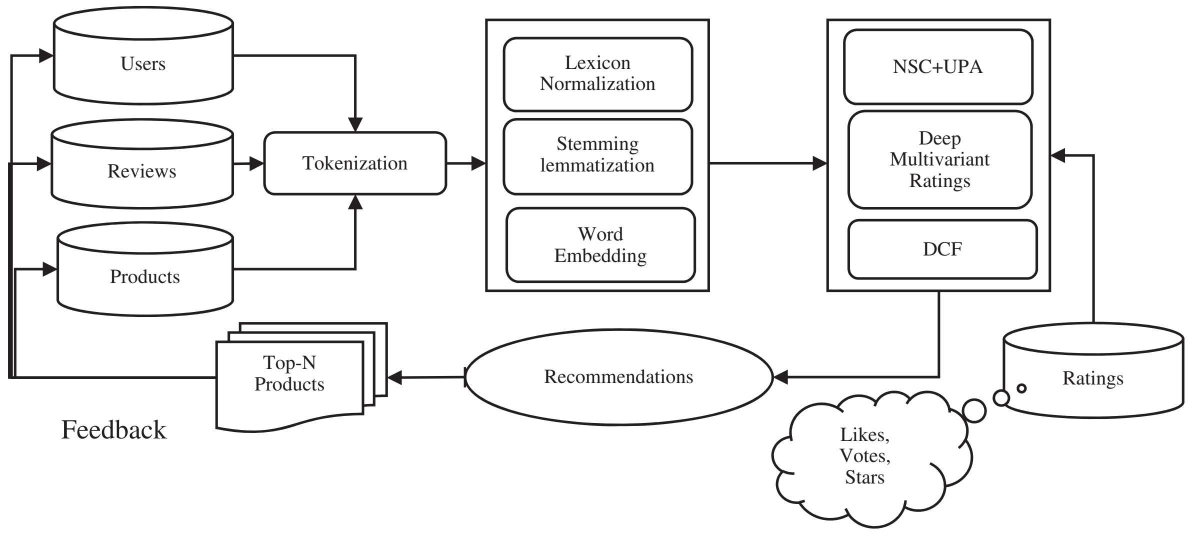

Figure 1: Model design

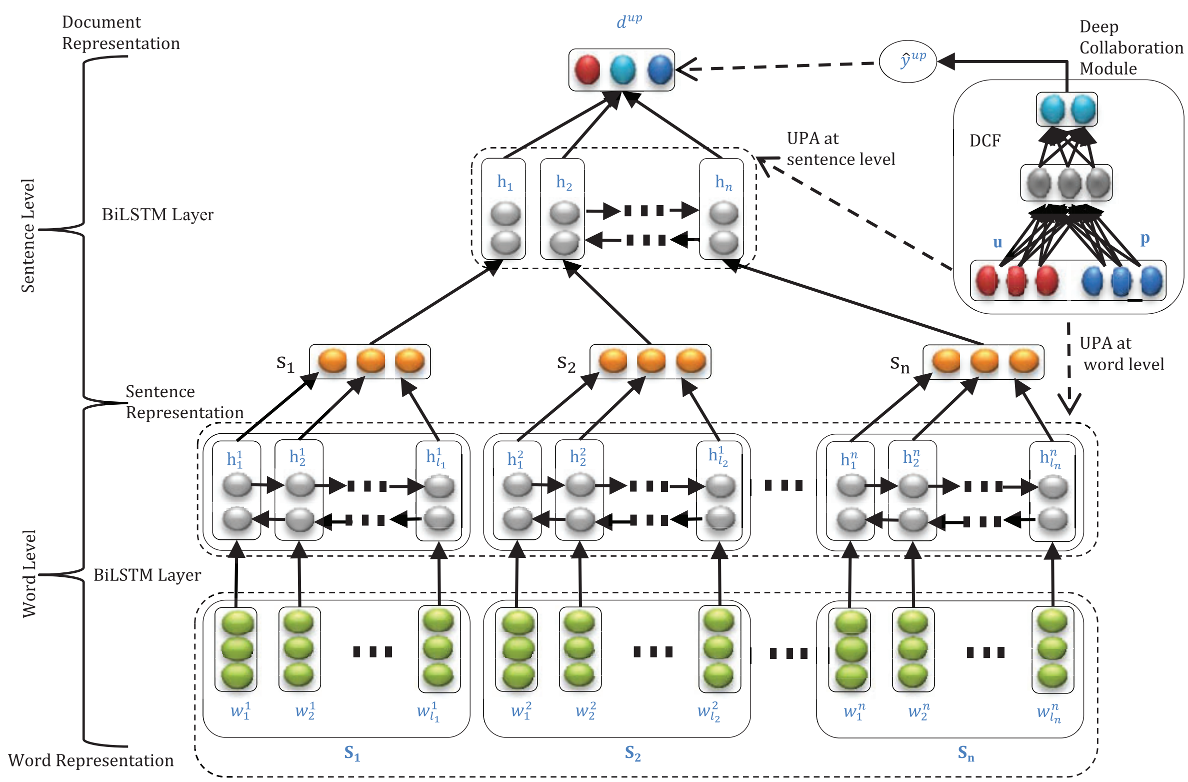

Figure 2: Architecture of HDCF

3.1.1 *Word Level Representation*

At the world level, each word is taken from the sentence S_i and mapped into the lower space of the dimension. This means that each word w_j^i is represented in embedding as w_j^i, which is mapped into the embedding as $w_j^i \in \mathbb{R}^d$. Each word $\mathrm{w}_\mathrm{j}^\mathrm{i}$ contains two states which are cell state and hidden state. The cell state is represented by c_j^i and hidden cell state is represented by h_j^i. These two states are updated by their represented previous states which are givens as $\mathrm{c}_{\mathrm{j-1}}^{\mathrm{i}}$ and $\mathrm{h}_{\mathrm{j-1}}^{\mathrm{i}}$. The equation for these states is given below:

$$\begin{bmatrix} i_j^i \\ f_j^i \\ o_j^i \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \end{bmatrix} \left(W.\left[h_{j-1}^i,\ w_j^i\right] + b\right) \tag{1}$$

$$\hat{c}_j^i = \tanh\left(W.\left(h_{j-1}^i, w_j^i\right] + b\right) \tag{2}$$

$$c_j^i = f_j^i \odot c_{j-1}^i + i_j^i \odot \hat{c}_j^i \tag{3}$$

$$h_{j=}^i\ o_j^i \odot \tanh\left(c_j^i\right) \tag{4}$$

In the above Eqs. (1)–(4), the variables $i,\ f,\ o$ are used for gate multiplication and $\odot$ is used for multiplication element-wise. The sigma in the equation is the sigmoid function while b and W are the factors that are selected for training. To get the representation of the sentence s_i or we need to feed the layers of $(h_1^i, h_2^i, \ldots, h_{l_i}^i)$ up to an average threshold level.

3.1.2 Sentence Level Representation

The sentence-level sentiment is the same as the word-level sentiment except for the difference that the feeding is done on the sentence except for the hidden layers. The feeding is done to the defined layer of the pool as per the same criteria of the word levels. The sentence feeding is done by the sentences represented by $(s_1,\ s_2, \ldots, s_n)$ to achieve the document d. The embedding process is done by LTSM.

$$\begin{bmatrix} i_i \\ f_i \\ o_i \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \end{bmatrix} (W.[h_{i-1},\ s_i] + b) \tag{5}$$

$$\widetilde{c}_i = \tanh(W.(h_{i-1},\ s_i] + b) \tag{6}$$

$$c_i = f_i \odot c_{i-1} + i_i \odot \widetilde{c}_i \tag{7}$$

$$h_i = o_i \odot \tanh(c_i) \tag{8}$$

Training parameters need to train are represented as $s_i,\ b_i$. The feed hidden states $(h_1,\ h_2, \ldots, h_n)$ is represented by a mediocre pooling layer to acquire the d_i document-representation shown in Eqs. (5)–(8).

3.1.3 User Product Attention Mechanism

The *User Product Attention* (UPA) is developed to gather the user's views about the crucial features related to the classification of semantics. The classification is imposed at different levels. The sentiment starts from the lowest level to classify the word level classification for the representation of sentences. Furthermore, sentence-level classification through the sentiment is done to achieve the representation of documents. The user product attention model is done from the lowest to the highest model in this way. It is confirmed that each word has a different perspective and meaning in different situations. So, the words are extracted from the gathered resources through the UPA model instead of the hidden layer feeding method of pooling. All the required words are extracted in this way. In the final step, the combined aggregate of the words is found. Through this aggregate, the sentence representation is formed. The representation of sentences is equal to the weighted addition of the hidden layers. The equation is given below:

$$s_i = \sum_{j=1}^{l_i} a_j^i h_j^i \tag{9}$$

$$d_i = \sum_{i=1}^{n} a_i h_i \tag{10}$$

The above Eqs. (9) and (10) are used to find the importance of any word in the occurrence of any specific sentience based on the hidden layer. For importance, a variable is used. The product p and user u are considered continuous variables $u \in \mathbb{R}^{d^u}$ and $p \in \mathbb{R}^{d^p}$ respectively. The embedding in the low dimension of both user and the product is shown through d^u and d^p. The weighted values of the attention for any specific word such as a_j^i is given by the following Eqs. (11) and (12).

$$a_j^i = \frac{\exp\left(e\left(h_j^i, u, p\right)\right)}{\sum_{k=1}^{l_i} \exp\left(e\left(h_k^i, u, p\right)\right)} \tag{11}$$

For sentence level

$$a_i = \frac{\exp\left(e\left(h_i, u, p\right)\right)}{\sum_{i=1}^{n} \exp\left(e\left(h_i, u, p\right)\right)} \tag{12}$$

In the above Eqs. (11) and (12), e shows the score of importance for any word in the specific sentence and document. The function of this variable e is defined in Eqs. (13) and (14) given below:

$$e\left(h_j^i, u, p\right) = v^T \tanh\left(W_H h_{ij} + W_U u + W_P p + b\right) \tag{13}$$

For sentence level

$$e\left(h_i, u, p\right) = v^T \tanh\left(W_H h_i + W_U u + W_P p + b\right) \tag{14}$$

For finding the e score weighted matrix of each defined variable is required. The weight matrix of the product is given by W_P, the weight matrix of the user is given by W_U and the weight matrix of the hidden layer is shown by W_H. The vector of the weight is given by v and the transpose of the v vector is given by v^T. The user product attention model UPA is also used at the sentence level because each sentence may vary in the different documents. The UPA for the sentence is found by extractive the important sentence in the document for the document representation purpose. The Eq. (15) of the sentence level UPA is given below:

$$s_i = \sum_{j=1}^{l_i} a_j^i h_j^i \tag{15}$$

At sentence level

$$d^{up} = \sum_{i=1}^{n} \beta_i h_i \tag{16}$$

where β is used as the weighted sum of the hidden value at the sentence level, therefore, the UPA is achieved as per the same criteria found in word-level sentiment analysis shown in Eq. (16).

3.2 Document Classification

The classification is the core of this research work as it changes the raw facts into meaning classes. The classes belong to different dimensions with different attributes. As in this research, work sentiment is done through the hierarchal level from words to the document d^{up}. And for the classification, the highest-level classification is done. The document d^{up} is represented as the feature for classification and documents are classified into n number of classes C with the non-linear method. The classification is done by the following Eq. (17).

$$\hat{d}^{up} = \mathrm{Tanh}\left(\mathrm{W}_c d^{up} + \mathrm{b}_c\right) \tag{17}$$

Tanh () activation function is used at an absolute layer to get the sentiment distribution of the document.

$$p_c = \frac{\exp\left(\hat{d}_c^{up}\right)}{\sum_{k=1}^{C} \exp\left(\hat{d}_k^{up}\right)} \tag{18}$$

where is p_c is the prediction probability of the class C in this research work shown in Eq. (18). While the classification error can occur the classification errors may result in the entropy of the gold sentiment function so the loss can be found in Eq. (19) given below:

$$L = \sum_{d^{up} \epsilon D} \sum_{c=1}^{C} p_c^g \left(d^{up}\right) . \log\left(p_c\left(d^{up}\right)\right) \tag{19}$$

In the above equation d^{up} is used for such documents that are used for training. The probability of gold sentiment is found by p_c^g and c is used to represent the grounded truth values which are 1 for true and 0 otherwise.

3.3 Neural Sentiment Classification

The process of identification and classification of opinions is stated in the microblog or short text to determine the topic, polarity, attitude, and emotions is known as sentiment analysis. Preprocessing and Text structuring is necessary for machine learning because the machine directly cannot understand the semantics of the text. Sentiment labeling of review is predicted by document-level classification. For sentiment analysis of reviews given by users for products, all words or sentences are not participating equally. Because some words or sentences solidly indicate the user's preferences while others show the product's characteristics. So, the sentiment label of reviews is inferred by two kinds of information with latent semantic representation in the user's review and product reviews by incorporating the information of user and product for NSC using a hierarchical user attention network and a hierarchical product attention network. Finally, the sentiment label of the review is predicted by the SoftMax classifier in which d^{up} is taken as a feature of users and products shown in Eqs. (20) and (21). Specifically, C classes of sentiment distribution for review, the SoftMax layer, and the Linear layer are implemented to project review representation d^{up}.

$$\hat{d}^{up} = \mathrm{SoftMax}\left(\mathrm{W}^{up} d^{up} + \mathrm{b}^{up}\right) \tag{20}$$

$$p_c = \frac{\exp\left(\hat{d}_c^{up}\right)}{\sum_{k=1}^{C} \exp\left(\hat{d}_k^{up}\right)} \tag{21}$$

3.4 Deep Multivariant Rating

Deep Multivariant Rating (DMR) is computed by the integration of continuous rating (sentiment of product reviews) and discrete (Stars, votes, and likes) to find out the true popularity of j^{th} product for true popularity. Here S^{p_i} is the normalized total stars rating about ith product, V^{p_i} is the normalized total votes rating about ith product and L^{p_i} is the normalized total likes rating about ith product given by different users. Here in Eq. (22) $C_{u_j}^{p_i}$ is a class favorability of user j for the product i.

$$r_{u_j}^{p_i} = \left[f\left(S^{p_i}\right); f\left(V^{p_i}\right); f\left(L^{p_i}\right); f\left(C_{u_j}^{p_i}\right)\right] \tag{22}$$

here $r_{u_j}^{p_i}$ is the concatenated multivariate overall rating of ith product for jth user. This popularity status will be determined by the multivariate value that will be measured by the integration of ratings (Likes, Votes, Stars, and the Sentiment score of reviews) about a product shown in Eqs. (23) and (24).

$$\hat{r}_{u_i}^{p_i} = \text{SoftMax}\left(\text{W}^{up} r_{u_i}^{p_i} + \text{b}^{up}\right) \tag{23}$$

$$p_c = \frac{\exp\left(\hat{r}_c^{up}\right)}{\sum_{k=1}^{C} \exp\left(\hat{r}_k^{up}\right)} \tag{24}$$

here p_c is used for a multivariant rating class.

3.5 Deep Collaborative Filtering

We implement the intermediate hidden layers shown in Eqs. (25) and (26) to design the user-product interactions module by feeding the d^{up} to obtain a non-linear complex relationship of user-product:

$$\begin{aligned}
z_1^{\text{up}} &= f_1^{\text{up}}\left(\text{W}_1^{\text{up}}.d_1^{\text{up}} + b_1^{\text{up}}\right) = \left(d^{up}\right) \\
z_2^{\text{up}} &= f_2^{\text{up}}\left(\text{W}_2^{\text{up}}.z_1^{\text{up}} + b_2^{\text{up}}\right) \\
z_3^{\text{up}} &= f_3^{\text{up}}\left(\text{W}_3^{\text{up}}.z_2^{\text{up}} + b_3^{\text{up}}\right) \\
&\ldots \\
z_L^{\text{up}} &= f_L^{\text{up}}\left(\text{W}_L^{\text{up}}.z_{L-1}^{\text{up}} + b_L^{\text{up}}\right)
\end{aligned} \tag{25}$$

$$\begin{aligned}
\hat{y}_{up} &= \sigma\left(\text{h}^T\left(z_L^{\text{up}}\right)\right) \\
\hat{y}_{up} &= tanh\left(\text{h}^T\left(z_L^{\text{up}}\right)\right)
\end{aligned} \tag{26}$$

here f_L^{up} the "*Activation Function*", W_L^{up} is "*Weight Matrix*" and b_L^{up} is "*Bias Vector*" is represented for the L^{th} layer while L represents the number of layers in the deep interaction module. Activation function as tanh () is applied. The interaction's output is the projected interaction value between the user u and the product p. Filtering is performed according to the user's profile or product profile using similarity measures. These features of the profile will be matched using cosine similarity techniques. "*Cosine similarity*" measurement is based on two vectors used for measuring the angle value of user profile and product profile genre for similarity manipulation and correlations are computed. Here zero angle represents the most similar to users' profile or product's profile and vice versa in Eqs. (27), (28), and (29). Here y_{up} and $\hat{y}_{up}$ maybe a user or product.

$$\overrightarrow{y_{up}}.\overrightarrow{\hat{y}_{up}} = \left\|\overrightarrow{y_{up}}\right\| . \left\|\overrightarrow{\hat{y}_{up}}\right\| . \cos\theta \tag{27}$$

$$sim\left(\overrightarrow{y_{up}}.\overrightarrow{\hat{y}_{up}}\right) = \cos\theta = \frac{\overrightarrow{y_{up}}.\overrightarrow{\hat{y}_{up}}}{\left\|\overrightarrow{y_{up}}\right\| . \left\|\overrightarrow{\hat{y}_{up}}\right\|} \tag{28}$$

$$top\_n\_\hat{y}_{up} = \hat{y}_{up} * popularity = \left[(1) + \frac{1}{(1) + \sqrt{r_{uj}^{p_i} *}} \right] \times \propto \quad (29)$$

The popularity status can be measured by using the multivariate overall rating at ten scales when $\propto = 10$.

4 Models Evaluation

These four famous *IMDb*[2], *Metacritic*[3] and "*Rotten Tomatoes*"[4] and "*Fandango*"[5] Product_ratings_2016_17[6] datasets are discussed and shown in fig 9.8 to raise the issues that lead to adopting the multivariant rating system. Yet how are these four platforms different, decrease the confidence, and are you supposed to trust to get the rundown on products? Here's what you need to say. So, we have now looked at what each of these platforms, along with their benefits and drawbacks, has to deliver. There's no one site, as you have already suspected, that is perfect for everything.

However, for various reasons, we discuss each of these sites. IMDb is a great place to see what the general public thinks about a product. If you don't know what the critics are saying and want to see what people like you think of a product, then use IMDb. Only be aware that viewers sometimes distort the vote at 10-star ratings, which can somewhat reduce scores. Rotten Tomatoes delivers the best overall impression on whether a product is worth watching in one instant. If you just trust high critics 'opinions and want to know if a product is at least good, then you can use Rotten Tomatoes. Although the Fresh/Rotten binary can oversimplify critics 'sometimes nuanced opinions, it can still assist you in weeding out junk products. The balanced aggregate score is provided by Metacritic. If you don't know which views of the reviewers go into the final score and want to see a general average, then use Metacritic. Its expectations are still unknown but Metacritic makes it easy to compare side-by-side professional reviews with feedback. Fandango has a consumer rating of 1.1 stars from 1,039 reviews indicating that most consumers are generally dissatisfied with their purchases. The confusion is caused by a uniform distribution and it was found that the quantitative ratings on Fandango's site were always rounded to the next largest half-star, rather than the closest one (for example, a 3.1 average rating for a movie should have been rounded to 3.5 stars, instead of 3.0). Here the following Fig. 3 represents the differences in Product ratings 2016-17 datasets.

We analyze the datasets and take 10 samples of the product to show how different ratings from different sites for the same product or products are varied. All problems in the discussion and graph show that different sites have different rating criteria and rating scores are different for the same product which reduces the confidence of users in the recommendation system and reduces the popularity of actual popular products or products. That is the reason we combine the discreet and continuous rating to generate the multivariant rating that incorporates the true rank product.

In [35] just only uses the implicate and explicate rating, no UPA is used, and deep learning is not used to find out the user's preference or taste of a movie from reviews or microblogs, in other words, he did not utilize the qualitative data. In the model multivariate rating is not used, while just using the [22] movie feature rating. Here the following Tables 2 and 3 represent the consequences of the recommender system.

[2] https://help.imdb.com/article/imdb/track-products-tv/ratings-faq/G67Y87TFYYP6TWAV#ratings
[3] https://www.metacritic.com/about-metascores
[4] https://www.rottentomatoes.com/critics/
[5] https://www.fandango.com/product-reviews
[6] https://github.com/mircealex/Product_ratings_2016_17

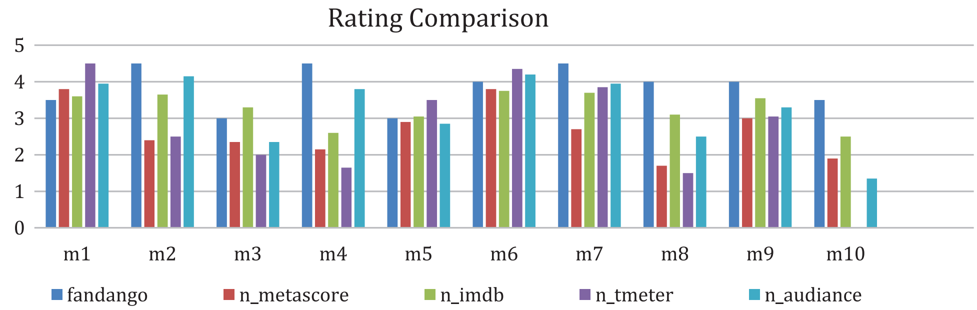

Figure 3: Product ratings 2016–17 differences

Table 2: Comparisons between different recommendation models

Models	FN	TN	FP	TP	Correct predicted	Incorrect predicted
[20] NCF	215	237	207	341	578	422
[26] J-NFC	175	355	95	375	730	270
[9] TFIDF + MR	131	347	116	406	753	247
[22] DeepCoNN	135	250	90	525	775	225
[10] NSC + MR	57	387	114	442	829	171
HDCF	5	433	8	554	987	13

Table 3: Comparisons between different recommendation models

Models	Precision	Recall	RMSE	Accuracy	F1
[20] NCF	0.6223	0.6133	13.34481	0.578	0.6178
[26] J-NFC	0.7979	0.6818	8.53815	0.73	0.7353
[9] TFIDF + MR	0.7778	0.7561	7.810826	0.753	0.7668
[22] DeepCoNN	0.8537	0.7955	7.115125	0.775	0.8235
[10] NSC + MR	0.7950	0.8858	5.407495	0.829	0.8379
HDCF	0.9858	0.9911	0.411096	0.987	0.9884

The results of sample data are applied to recommendation models. The HDFC gained an accuracy near about 98.70% with 0.4111 RMSE. The true positive rate of HDFC is 0.99106, which means the system recommended movies truly interesting to users, and the false positive rate is 0.01814, which means the system does not recommend movies by HDFC to be truly not interesting to users. The [20] NCF that don't user-product attention mechanism while [26] J-NFC uses the User Product Attention jointly which improves accuracy as compared to NCF [9]. TFIDF + MR uses the TFIDF weighting scheme for sentiment scoring as well as using the multivariant rating that increases the accuracy [22]. DeepCoNN provides better results as compared to the previous [10]. NSC + MR combined the Neural sentiment classifier with a Multivariant rating to improve the accuracy as compared previously addressed. HDCF combined the NCF + UPA + NSC + DMR gained more accuracy as compared to baseline models.

The below Fig. 4 justifies our approach with the other's models. The reasons are that just determining the polarity of the term is not enough to evaluate the reviews for significant recommendation and in [22] DeepCoNN was used to determine the user and movie information, while HDFC used NSC-UPA to evaluate the sentiment score of reviews as well as useful product interaction for Deep Collaboration in the recommendation.

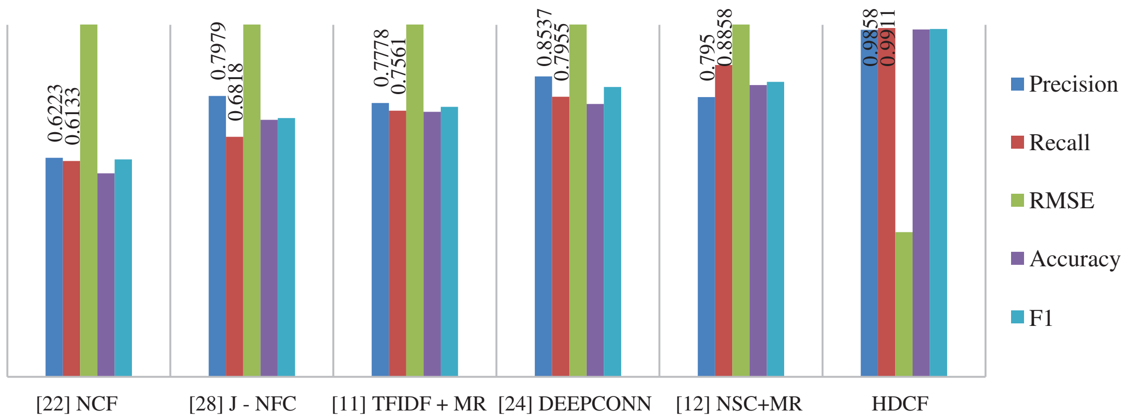

Figure 4: Differences in accuracy and RMSE of different decision parameters and recommendation models

In movie recommendation systems, Suggestions and Predictions are done on the bases of just bi-ratings parameters or uni-rating parameters. Furthermore, sentiment classification or rating to determine the movie popularity is not reliable and significant, these models could not provide better recommendation services, because there is a huge gap between statistical information of discrete ratings (Stars, Votes, and Likes) and continues ratings. Because some of them are producing too much high popularity and another's too low popularity of the same movie. Reference [9] gives the novel idea for multivariant ratings but in which deep learning is not implemented, in which the TFIDF approach is used to evaluate weights of terms for determining the sentiment of movie reviews as well as machine learning is implemented for classification. Therefore, significant values are required, and semantic information is required in an efficient way using LSTM-UPA deep learning algorithm for better semantic analysis and user movie interaction, so true semantic value increases the significance of trust in the recommendation system. Sentiment classification is improved by using UPA at word level and sentence level with average pooling word and sentence level to improve the semantic information about reviews product. The reason for applying the attention at the word level is to improve the semantic information of the document as compared to just only applying it at the sentence level.

Here following Fig. 4 represents the Accuracy and RMSE of HDCF with baseline models.

5 Conclusion and Future Work

Through this research work, we investigated the collaborative filtering of neural network architectures. This work complements the standard shallow models for collaborative filtering, opening up a new field of recommendation-based research opportunities based on deep learning for the medical domain to extract the history of COVID-19 for treatment. For the recommender system we propose a novel model of HDCF that contains the four major modules that are jointly and tightly coupled. initially, user and product features are learned and encoded through UPA to incorporate the user's preferences and product characteristics, and user-product interaction learning through multi-layer perceptron for determining the *User Product Interaction* (UPI). Using the feature vectors of users and products as inputs fed to the user-product interactions module for the prediction of users or product that has a similar taste to other by measuring the similarity of users or products in an improved and reliable way.

There is a limitation to restricting the model, high data availability, multiple data source for the same product and users, and high computations, that leads to better evaluation.

This research is based on sentiment classification with the help of neural sentiment classification. The sentimental results of the implemented models show that it performs with high accuracy concerning other models. To make HDCF suitable for the top-N recommendation task, we add a revolutionary feature for losses. This requires data from both. First, concerning the research study, we propose HDCF with more auxiliary information which covers the research of [20] and [26]. It is also essential to analyze heterogeneous information in an information base to enhance the efficiency of deep learning recommending systems. Secondly, in a session with HAN, we explore a user's contextual information to discuss complex aspects of systems. Additionally, a system of focus applied to HDCF, could filter out uninformative material and choose the most relevant products while at the same time maintaining better understanding. The findings from the studies indicate the HDCF's effectiveness. We also experimentally examined the performance of HDCF under circumstances, e.g., besides, we also evaluated the HDCF model with a large dataset, i.e., the results indicate that HDCF also gives outstanding performance against state-of-the-art baseline models on given datasets. Finally, because we found HDCF to be computationally more expensive than NCF, we expect to refine our model's structure and details of implementation to increase the performance.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] L. Jiang, Y. Cheng, L. Yang, J. Li, H. Yan *et al.,* "A trust-based collaborative filtering algorithm for E-commerce recommendation system," *J. Ambient Intell. Humaniz. Comput.*, 2019, https://doi.org/10.1007/s12652-018-0928-7.

[2] C. Bin, T. Gu, Y. Sun, L. Chang, W. Sun *et al.,* "Personalized POIs travel route recommendation system based on tourism big data," 2018. https://doi.org/10.1007/978-3-319-97310-4_33.

[3] A. K. Sahoo, C. Pradhan, R. K. Barik, and H. Dubey, "DeepReco: Deep learning based health recommender system using collaborative filtering," *Computation*, 2019, https://doi.org/10.3390/computation7020025.

[4] P. Chen and J. Li, "A recurrent model with self-attention for product repurchase recommendation," 2019. https://doi.org/10.1145/3325730.3325763.

[5] Q. Yao, X. Liao, and H. Jin, "Hierarchical attention based recurrent neural network framework for mobile MOBA game recommender systems," 2019. https://doi.org/10.1109/BDCloud.2018.00060.

[6] R. Huang, S. McIntyre, M. Song, E. Haihong, and Z. Ou, "An attention-based recommender system to predict contextual intent based on choice histories across and within sessions," *Appl. Sci.*, 2018, https://doi.org/10.3390/app8122426.

[7] V. P. Khadse, A. P., S. M. Basha, N. C. S. N. Iyengar, and R. D. Caytiles, "Recommendation engine for predicting best rated movies," *Int. J. Adv. Sci. Technol.*, 2018, https://doi.org/10.14257/ijast.2018.110.07.

[8] A. Da'U and N. Salim, "Sentiment-aware deep recommender system with neural attention networks," *IEEE Access*, 2019, https://doi.org/10.1109/ACCESS.2019.2907729.

[9] M. Ibrahim and I. S. Bajwa, "Design and application of a multi-variant expert system using apache hadoop framework," *Sustain.*, 2018, https://doi.org/10.3390/su10114280.

[10] M. Ibrahim, I. S. Bajwa, R. Ul-Amin, and B. Kasi, "A neural network-inspired approach for improved and true movie recommendations," *Comput. Intell. Neurosci.*, 2019, https://doi.org/10.1155/2019/4589060.

[11] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, 2005. https://doi.org/10.1109/TKDE.2005.99.

[12] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys*, 2019. https://doi.org/10.1145/3285029.

[13] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer (Long. Beach. Calif).*, 2009, https://doi.org/10.1109/MC.2009.263.

[14] B. T. Betru, C. A. Onana, and B. Bernabe, "Deep learning methods on recommender system: A survey of state-of-the-art," *Int. J. Comput. Appl.*, 2017, https://doi.org/10.5120/ijca2017913361.

[15] L. W. Huang, B. T. Jiang, S. Y. Lv, Y. B. Liu, and D. Y. Li, "Survey on deep learning based recommender systems," *Jisuanji Xuebao/Chinese Journal of Computers*, 2018. https://doi.org/10.11897/SP.J.1016.2018.01619.

[16] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," 2016. https://doi.org/10.1145/2959100.2959165.

[17] H. Wang, N. Wang, and D. Y. Yeung, "Collaborative deep learning for recommender systems," 2015. https://doi.org/10.1145/2783258.2783273.

[18] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," 2007. https://doi.org/10.1145/1273496.1273596.

[19] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-N recommender systems," 2016. https://doi.org/10.1145/2835776.2835837.

[20] X. He, L. Liao, H. Zhang, L. Nie, X. Hu *et al.,* "Neural collaborative filtering," 2017. https://doi.org/10.1145/3038912.3052569.

[21] H. J. Xue, X. Y. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," 2017. https://doi.org/10.24963/ijcai.2017/447.

[22] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," 2017. https://doi.org/10.1145/3018661.3018665.

[23] T. Bansal, D. Belanger, and A. McCallum, "Ask the GRU: Multi-task learning for deep text recommendations," 2016. https://doi.org/10.1145/2959100.2959180.

[24] D. Tang, B. Qin, and T. Liu, "Learning semantic representations of users and products for document level sentiment classification," 2015. https://doi.org/10.3115/v1/p15-1098.

[25] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, "Neural sentiment classification with user and product attention," 2016. https://doi.org/10.18653/v1/d16-1171.

[26] W. Chen, F. Cai, H. Chen, and M. D. E. Rijke, "Joint neural collaborative filtering for recommender systems," *ACM Trans. Inf. Syst.*, 2019, https://doi.org/10.1145/3343117.

[27] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola *et al.,* "Hierarchical attention networks for document classification," 2016. https://doi.org/10.18653/v1/n16-1174.

[28] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-a case study," *ACM WebKDD 2000 Web Min. ECommerce Work.*, 2000, https://doi.org/10.3141/1625-22.

[29] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," 2015. https://doi.org/10.1145/2806416.2806527.

[30] S. Sedhain, A. K. Menony, S. Sannery, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," 2015. https://doi.org/10.1145/2740908.2742726.

[31] J. Lian, F. Zhang, X. Xie, and G. Sun, "CCCFNet: A content-boosted collaborative filtering neural network for cross domain recommender systems," 2019. https://doi.org/10.1145/3041021.3054207.

[32] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," 2017. https://doi.org/10.24963/ijcai.2017/239.

[33] N. Sivaramakrishnan, V. Subramaniyaswamy, A. Viloria, V. Vijayakumar, and N. Senthilselvan, "A deep learning-based hybrid model for recommendation generation and ranking," *Neural Comput. Appl.*, vol. 33, no. 17, 2021, https://doi.org/10.1007/s00521-020-04844-4.

[34] R. Wang, Z. Wu, J. Lou, and Y. Jiang, "Attention-based dynamic user modeling and deep collaborative filtering recommendation," *Expert Syst. Appl.*, vol. 188, no. March 2021, pp. 116036, 2022, https://doi.org/10.1016/j.eswa.2021.116036.

[35] M. Y. Hsieh, W. K. Chou, and K. C. Li, "Building a mobile movie recommendation service by user rating and APP usage with linked data on hadoop," *Multimed. Tools Appl.*, 2017, https://doi.org/10.1007/s11042-016-3833-0.