

# A Hybrid Deep Collaborative Filtering Approach for Recommender Systems

Kirubahari R (✉ [kirubahari.cse2017@gmail.com](mailto:kirubahari.cse2017@gmail.com))

K.L.N. College of Engineering

Miruna Joe Amali S

K.L.N. College of Engineering

---

## Research Article

**Keywords:** Personalized Recommendation, Collaborative Filtering, Parallel Hybrid Recommender System, Cold start problem, Deep Neural Network, Matrix Decomposition

**Posted Date:** June 29th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-651522/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

# **A Hybrid Deep Collaborative Filtering Approach for Recommender Systems**

**R. Kirubahari <sup>a,\*</sup>, Dr.S. Miruna Joe Amali <sup>a</sup>**

**<sup>a</sup> Department of Computer Science and Engineering**

**K.L.N. College of Engineering, Pottapalayam,**

**Sivagangai District, Tamilnadu, India.**

**\*Corresponding author: Email address: [kirubahari.cse2017@gmail.com](mailto:kirubahari.cse2017@gmail.com)**

## **Abstract**

Recommender Systems (RS) help the users by showing better products and relevant items efficiently based on their likings and historical interactions with other users and items. Collaborative filtering is one of the most powerful technique of recommender system and provides personalized recommendation for users by prediction rating approach. Many Recommender Systems generally model only based on user implicit feedback, though it is too challenging to build RS. Conventional Collaborative Filtering (CF) techniques such as matrix decomposition, which is a linear combination of user rating for an item with latent features of user preferences, but have limited learning capacity. Additionally, it has been suffering from data sparsity and cold start problem due to insufficient data. In order to overcome these problems, an integration of conventional collaborative filtering with deep neural networks is proposed. A Weighted Parallel Deep Hybrid Collaborative Filtering based on Singular Value Decomposition (SVD) and Restricted Boltzmann Machine (RBM) is proposed for significant improvement. In this approach a user-item relationship matrix with explicit ratings is constructed. The user - item matrix is integrated to Singular Value Decomposition (SVD) that decomposes the matrix into the best lower rank approximation of the original matrix. Secondly the user-item matrix is embedded into deep neural network model called Restricted Boltzmann

Machine (RBM) for learning latent features of user- item matrix to predict user preferences. Thus, the Weighted Parallel Deep Hybrid RS uses additional attributes of user - item matrix to alleviate the cold start problem. The proposed method is verified using two different movie lens datasets namely, MovieLens 100K and MovieLens of 1M and evaluated using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The results indicate better prediction compared to other techniques in terms of accuracy.

**Keywords:** Personalized Recommendation, Collaborative Filtering, Parallel Hybrid Recommender System, Cold start problem, Deep Neural Network, Matrix Decomposition

## **1. Introduction**

Recommender system (RS) is one of the most powerful decision-making process. With fast growth of information in internet creates the information overload problem. To address this issue, several recommender systems (RS) has been developed to help users to find better products <sup>21</sup>. These recommender systems are used for providing personalized recommendations based on user profile and user behaviours' <sup>4</sup>. These systems have been used in various products and services on the internet to help the user find and select the items like books, movies, foods, music, etc. Many web services like Amazon, Netflix, YouTube and other social networks have adopted to recommender systems. The use of recommendation system in music to discover new songs for users <sup>18</sup> and like-minded friends in social network <sup>20</sup> is also available in literature.

The input to a recommender system is generally the user rating (numerical value), demographic data (age, gender, occupation), content data which are textual analysis of related items rated by user and also depends on the type of filtering algorithm like Collaborative Filtering (CF), Content-Based Filtering (CBF) and Demographic Filtering (DF) used for RS. The output of recommender systems can be either prediction or a top recommendation. A prediction is a numerical value  $r_{a,j}$ , which represents the anticipated opinion of

active user  $u_a$  where  $a=1,2,\dots,m$ , for item  $i_j$  where  $j=1,2,\dots,n$ . Recommendation is representing a list of top  $N$  items where  $N \leq n$ , which the active user is expected to like most.

Collaborative Filtering (CF) is one of the key techniques of personalized recommender systems<sup>6</sup>. This method is classified as memory-based and model-based. The main idea of Collaborative Filtering is that if two users have similar likings of certain items in the past, they will also like similar items in future. These recommendations are not only based on user past but also an users' explicit ratings and implicit rating. The traditional CF creates the user-item matrix based only on preferences for particular items given by users. Some limitations of CF are cold-start and data sparsity problems. The cold-start problem is related to recommendations for new user, new item or new community. In case of new users or new items arrives, the system does not have sufficient data to predict about their user / item preferences in order to make recommendations. If a new item does not have any user ratings, the user-item matrix is very sparse and the recommendation accuracy usually drops very obviously.

The matrix factorization (MF) is one of the most popular collaborative filtering techniques. The MF approach is one of the most accurate approaches to reduce the problem of high levels of data sparsity<sup>1</sup>. MF can map both users - items to a joint latent factor space of dimensionality, by factoring the user-item interaction matrix. Using matrix decomposition techniques, User-item interactions are showed as inner products in that space<sup>8</sup>. In latent space, based on similarities among the users and items the recommender system predicts a personalized ranking over a set of items for each individual user. So, this model finds the hidden structure behind the data accurately. The popular decomposition models are Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Probabilistic Matrix Factorization (PMF) and Non-Negative Matrix Factorization (NMF). MF invariably reduces the

dimension of data, therefore causes difficulty in improving the prediction accuracy. In addition, it also suffers from cold start problem and data sparsity issues. In order to address these issues, the idea is to build deep learning-based hybrid recommendation method.

Recently, a powerful representation learning approach called deep learning methods have been successfully applied to various fields such as Computer Vision, Image and Speech Recognition, Natural Language Processing and Medical Image Analysis. Most popular deep learning techniques are Convolutional Neural Networks (CNN), AutoEncoder (AE), Restricted Boltzmann Machine (RBM), and Recurrent Neural Network (RNN). Restricted Boltzmann Machine (RBM) was firstly proposed to model users' explicit ratings on items <sup>16</sup> in recommendation systems. AutoEncoder (AE) is neural network model attempting to reconstruct its input data in the output layers, which consists of encoder and decoder component <sup>19</sup>. There are many variants of autoencoders such as marginalized denoising autoencoder, sparse autoencoder, denoising autoencoder, contractive autoencoder and variational autoencoder. Autoencoder and denoising autoencoder have also been applied for recommendation <sup>10</sup>. These methods are used to reconstruct the users' ratings through learning hidden structures with the explicit historical ratings.

Nowadays researchers have explored designing collaborative filtering based Deep Neural Network (DNN) for recommendations <sup>21</sup>. However, traditional CF methods and deep learning train their models in a batch learning algorithm or train entire data set at once. The growth of information, continuous data streams are stored in the system from time to time. In each time the user interest and preferences are changing all the time. When new data arrives in the system, traditional CF and Deep learning have to train the model from scratch. These methods are expensive for them to retrain and update the model's parameters <sup>12</sup>.

The Convolutional Neural Networks (CNN) model is used to extract user-item features from users and items side information to resolve the new data arrival problem by deep bias probabilistic matrix factorization (DBPMF) model<sup>8</sup>. One of the major limitations in CF is data sparsity problem and it addressed by fuzzy interference rules to enhance the recommendation accuracy<sup>1</sup>. This model first categorizes the user like, user dislike and common users' like and dislike both and finally measures the loss at attribute at feature due to sparsity. The rule of interference plays a vital role at product rating and measures the correlation of recommendation matrix.

Many recommendation models focus on user ratings, user past behaviour, item information and user/item side information. This information cannot fully expose user-item relations for accurate prediction. Hybrid weighting is a hybridization technique that computes the predication score of all recommendation approaches by giving weight to each approach and summing these weights to produce a new output recommendation. In order to elevate cold start problem a Weighted Parallel Deep Hybrid Collaborative Filtering recommendation method is proposed.

The proposed work is summarized as follows:

- A user-item relationship matrix with explicit ratings is constructed. The user -item matrix is integrated to Singular Value Decomposition (SVD) that decomposes the matrix into the best lower rank approximation of the original matrix.
- Secondly embedding the user-item matrix into the deep neural network model called Restricted Boltzmann Machine (RBM) for learning latent features to predict user preferences.
- Finally, both SVD and RBM recommendation approaches gives their own prediction rating. A weighted hybridization combines these two values, by experimenting weight values  $W_1$  and  $W_2$  and generating the hybrid final prediction result.

Thus, the Weighted Parallel Deep Hybrid RS uses additional attributes of user - item matrix is to alleviate the cold start problem. Experimental results demonstrate that the proposed hybridization technique achieves better prediction accuracy compared to other techniques performance using movie lens dataset (MovieLens 100K and MovieLens of 1M).

## 2. Problem formulation

The goal of a recommender system is to recommend new items or to predict a certain item for a particular user based on the user's previous information. In a collaborative filtering strategy, there is a list of  $M$  users and  $N$  items with extremely sparse rating matrix  $R \in R^{M \times N}$ . Each entry  $R_{ij}$  represents the rating about the user  $i$ 's rating on item  $j$ . If  $R_{ij} \neq 0$  means the rating about the user  $i$  on the item  $j$  is observed, otherwise not observed.  $U \in R^{K \times M}$  and  $V \in R^{K \times N}$  denote the user and item latent factor vector respectively, where  $K$  is the dimensionality of the latent space. Moreover, the additional side information matrix of user and item is denoted by  $X \in R^{M \times p}$  where  $p$  is the side information of user and  $Y \in R^{N \times q}$  where  $q$  is the side information of item. Then, the corresponding matrix forms of latent factors for user and items are  $U = u_{1:m}$  and  $V = v_{1:n}$  respectively. Given the sparse rating matrix  $R$  and side information matrix  $X$  and  $Y$ , the goal is to learn user latent factors  $U$  and item latent factors  $V$  and hence to predict the missing ratings in  $R$ .

### 2.1 Matrix Factorization

Matrix Factorization (MF) map both users - items to a joint latent factor space of dimensionality, by factoring the user-item interaction matrix. Using matrix decomposition techniques, User- item interactions are showed as inner products in that space <sup>4</sup>. The MF decomposes the original rating matrix  $R$  into low-rank matrices  $U$  and  $V$  consisting of the user and item latent factor vectors respectively, such that  $R \approx UV$ . Given the latent factor vectors for users and

items, a user's rating for a movie is predicted by the inner product of those vectors.  $U \in R^{K \times M}$  and  $V \in R^{K \times N}$  denote the user and item latent factor vector respectively, where the dimensionality of the latent space is denoted as  $K$ . Accordingly, each user  $i$  is associated with a vector  $U_i \in R^K$ , and each item  $j$  is associated with a vector  $V_j \in R^K$ . For a given item  $j$ , the elements of  $V_j$  measure the degree to which the item possesses those factors as positive or negative. The resulting dot product  $V_j^T U_i$  captures the interaction between the user  $i$  and item  $j$ , the user's overall interest in the item characteristics. This approximates user  $i$ 's rating of item  $j$  which is denoted by  $r_{ij}$  leading to the estimate

$$r_{ij} = V_j^T U_i \quad (1)$$

To learn the factor vectors ( $U_i$  and  $V_j$ ), the system minimizes the regularized squared error on the set of known ratings as:

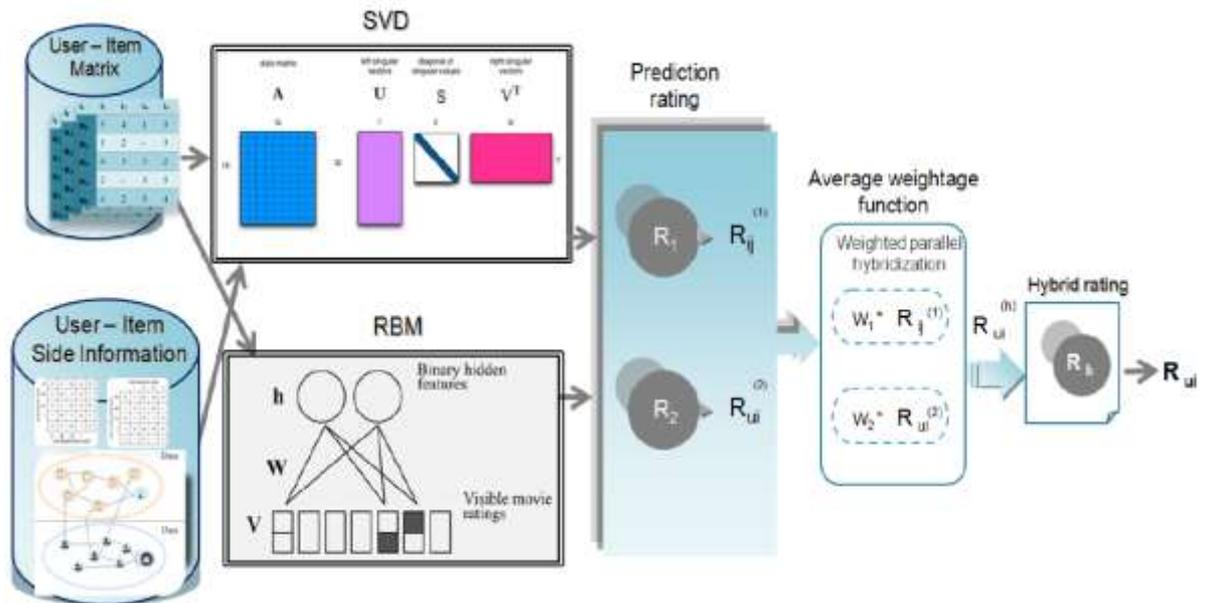
$$\min_{V^*, U^*} \sum_{(i,j) \in f} (r_{ij} - V_j^T U_i)^2 + \lambda (\|V_j\|^2 + \|U_i\|^2) \quad (2)$$

Here,  $f$  is the set of  $(i, j)$  pairs for which  $r_{ij}$  is known as the training set. The constant  $\lambda$  controls the extent of regularizations and is usually determined by cross-validation.

### 3. Proposed Hybrid Model

Hybrid recommendation model easily combine different algorithm that might do totally two different things by having a common interface on every recommender. There is more than one way to do a hybrid recommender system perhaps, to reserve some slots for top  $N$  recommenders and one recommender provides the accuracy for prediction. Also rating prediction from many recommenders in parallel and add average of the ratings together before ranging them. The proposed model is a Weighted Parallel Deep Hybrid RS, which is a combination of two technique called as collaborative filtering and deep learning technique. A weighted hybridization technique combines two or more

recommendation systems by computing weighed sums of their prediction rating scores. These scores are hybridized by using a uniform weighting scheme. It enhances the overall performance of the system.



**Figure. 1.** Architecture diagram of Proposed Hybrid Recommendation System

Figure.1 shows the architecture of a Weighted Parallel Deep Hybrid Collaborative Filtering technique for recommendation system. In this approach user-item matrix with explicit rating is given as input for both SVD and RBM method. In order to alleviate the cold-start problem of recommendation, also integrate user-item side information such as age, gender, occupation of users, release date and genres of movies as input for both SVD and RBM method. In recommendation result, both SVD and RBM algorithms have their own predication rating score  $R_{ij}$  and  $R_{ui}$ . By experimenting, weight values  $W_1$  and  $W_2$  are assigned to each algorithm for making final weighted average prediction score  $R_h$  for hybridization.

Collaborative filtering can predict accurately but the cold start problem exists in this technique. The SVD is a powerful technique of dimensionality reduction and to producing a low-rank approximation with excellent scalability

and accuracy. Computing the SVD is far more expensive than most of the other techniques. Singular value decomposition provides inferior solution for cold start problems. Hence, SVD alone is not suitable for recommendation. SVD provides a satisfactory solution to RS, it is less effective when new items do not accumulate enough data.

Deep learning is a machine learning method. Given a large dataset of input and output pairs, a deep learning algorithm will try to minimize the difference between its predication and expected output. RBM is the oldest approach of deep learning used for RS. RBM are neural network that belongs to energy-based model. It is probabilistic, unsupervised, generative deep machine learning algorithm. RBM can learn a probability distribution over its set of inputs and is a more efficient learning algorithm. RBM's objective is to find the joint probability distribution that maximizes the log-likelihood function. RBM is a non-deterministic approach, that is RBM nodes make stochastic decision to decide either to turned on or off an input. RBM is used in the domain of rating prediction for learning user preferences and item ratings by inferring latent features. The restriction in RBM is that, there is no intra-layer communication. This restriction allows for more efficient training algorithms. To build a robust generative model, the RBM performs several forward and backward passes to learn the optimal weights.

In recommendation result, both SVD and RBM algorithms have their own predication rating score. By experimenting, weight values  $W_1$  and  $W_2$  is assigned to each algorithm for making final weighted average prediction score for hybridization. This hybridization technique leverages their benefits for learning more expressive models. Weighted parallel hybridization can overcome the shortcoming of those two techniques when implemented stand alone. In order to cover the drawbacks of each technique with the advantages of other technique, both techniques are combined with an approach known as weighted parallel hybrid technique.

### 3.1 Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) is the powerful technique of dimensionality reduction. The key issue in SVD decomposition is producing a low-rank approximation. Given a  $m \times n$  matrix  $A$ , with rank  $r$ , the singular value decomposition  $SVD(A)$  is defined as

$$SVD(A) = U \times S \times V^T \quad (3)$$

Where  $U$  and  $V$  are orthogonal matrices with dimensions  $m \times m$  and  $n \times n$  respectively. Matrix  $S$  is a diagonal matrix also called singular matrix with dimension  $m \times n$  having non-negative real numbers. The set of initial  $r$  values of  $S$  ( $s_1, s_2, \dots, s_r$ ) are all positive with  $s_1 \geq s_2 \geq s_3, \dots, \geq s_r$ . The first  $r$  columns of  $U$  are eigen vectors of  $AA^T$  and represent the left singular vectors of  $A$ . Similarly, the first  $r$  columns of  $V$  are eigen vectors of  $A^T A$  and represent the right singular vectors of  $A$ .  $SVD$  provides the best low-rank approximation of the original matrix  $A$ . It is obtained by retaining the first  $k$  diagonal values of, by removing  $r - k$  columns from  $U$  and by removing  $r - k$  rows from  $V$ , which can be represented as follows:

$$A_k = U_k \times S_k \times V_k^T \quad (4)$$

The reconstructed matrix  $A_k$  is the closest approximation to the original matrix  $A$ . The best-  $k$  rank approximation of matrix  $A$  with respect to the Frobenius norm can be represented by:

$$\| A - A_k \|_F^2 = \sum_{i,j} (a_{ij} - \sum_k U_{ik} \times S_k \times V_{kj}^T)^2 \quad (5)$$

Prediction generation using  $SVD$ :

Once the  $m \times n$  ratings matrix  $R$  is decomposed and reduced into three  $SVD$  component matrices with  $k$  features  $U_k, S_k$  and  $V_k$ , prediction can be generated from it by computing the cosine similarities (dot products) between  $m$  pseudo-customers  $U_k \cdot \sqrt{S_k^T}$  and  $n$  pseudo-products  $S_k \cdot \sqrt{V_k^T}$ . In particular, the

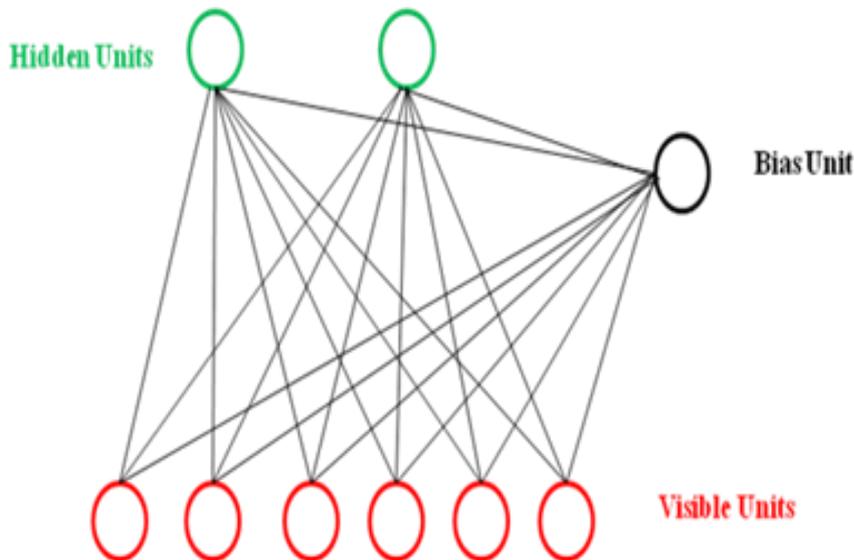
prediction score  $R_{i,j}$  for the  $i^{\text{th}}$  customer on the  $j^{\text{th}}$  product by adding the row average  $\bar{r}_i$  to the similarity.

$$R_{i,j} = \bar{r}_i + U_k \cdot \sqrt{S_k^T(i)} + S_k \cdot \sqrt{V_k^T(j)} \quad (6)$$

Once the *SVD* decomposition is done, the prediction generation process involves only a dot product computation, which takes  $O(1)$  time, since  $k$  is a constant.

### 3.2 Restricted Boltzmann Machines (RBM)

RBM is a two-layer stochastic neural network consisting of visible and hidden units. It has one layer of visible units (users' movie preference), one layer of hidden units (the latent factors) and a bias unit (whose state is always on, and adjust for the different inherent popularities of each movie) in Figure 2. Also, each visible unit is connected to all the hidden units in an undirected form, so each hidden unit is connected to all the visible units and each bias unit is connected to the all visible and hidden units. In order to make the learning easier, the network is restricted that there is no interconnection within visible layer and also hidden layer.



**Figure. 2.** A Framework of Restricted Boltzmann Machine

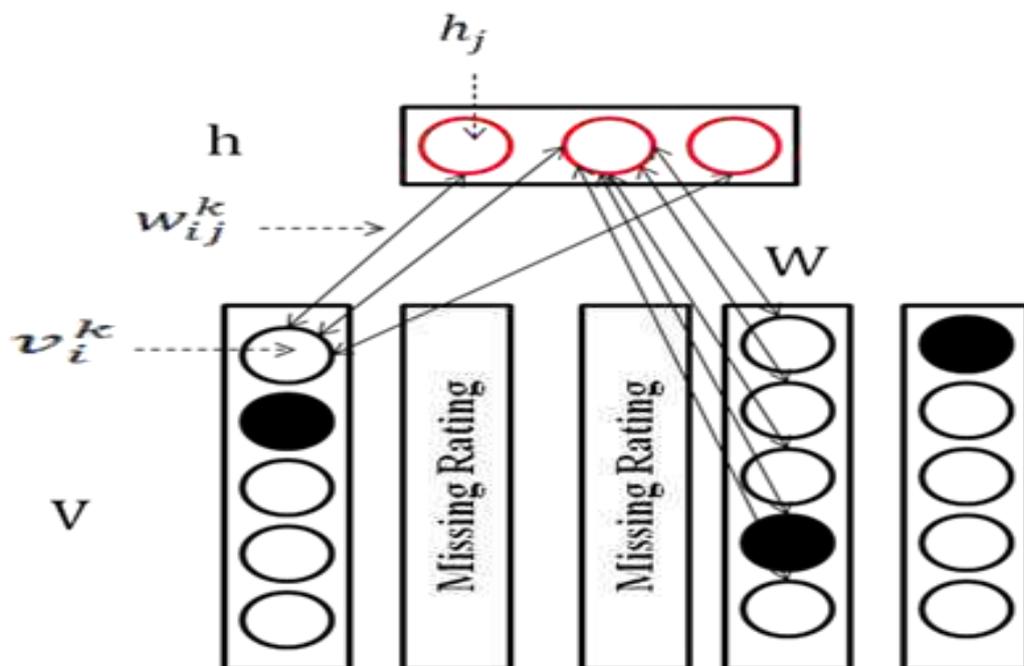
The stochastic, binary visible units encode user preferences on the items from the training data, therefore the state of every visible unit is known. Hidden

units are also stochastic, binary variables that capture the latent features. The network assigns a probability to each possible pair of a hidden and a visible vector via this energy function:

$$p(v, h) = \frac{e^{-E(v,h)}}{Z} \quad (7)$$

Where  $E$  is the energy of the system and  $Z$  is a normalizing factor as defined in <sup>5</sup>. To train for the weights, a contrastive divergence method was proposed by Hinton (Hinton, 2002).

Consider, there are  $M$  movies,  $N$  user and integer ratings on scale from 1 to  $K$ . The first problem in applying RBM to movie ratings is efficiently dealing with the large number of missing ratings. The solution proposed in <sup>16</sup> is to consider that the visible units corresponding to the movies that the user did not rate simply do not exist. In practice, these visible units are always turned off and hence their state is always zero. Alternatively, this can be seen as using a unique RBM per user, all sharing the same units but each only including the softmax units for the movies rated by their user.



**Figure. 3.** Restricted Boltzmann Machine for Collaborative Filtering

In Figure.3 binary visible units are replaced with softmax units. For each user, the RBM only includes the softmax units for the movies rated by that user. Let us consider some user  $u$  rated  $m$  movies. Let visible unit  $V$  be a  $K \times m$  matrix such that  $v_i^k = 1$  if  $u$  rated movie  $i$  as  $k$  and 0 otherwise. Let  $h_j, j=1, \dots, F$ , be the binary values of the hidden units of user features. Columns of  $V$  are modelled using a multinomial (a softmax) and hidden latent feature  $h$  are modelled just like in equation (6) and (7).

$$p(v_i^k = 1|h) = \frac{\exp(b_i^k + \sum_{j=1}^F h_j W_{ij}^k)}{\sum_{l=1}^K \exp(b_i^l + \sum_{j=1}^F h_j W_{ij}^l)} \quad (8)$$

$$p(h_j = 1|h) = \frac{1}{1 + \exp(-b_j - \sum_{i=1}^m \sum_{k=1}^K v_i^k W_{ij}^k)} \quad (9)$$

Where  $W_{ij}^k$  is the weight on the connection between the rating  $k$  of movie  $i$  and the hidden unit  $j$ ,  $b_i^k$  is the bias of rating  $k$  for movie  $i$  and  $b_j$  the bias term of hidden unit  $j$ .

The marginal distribution over the visible ratings  $V$  is equation (10):

$$p(V) = \sum_h \frac{\exp(-E(V, h))}{\sum_{V', h'} \exp(-E(V', h'))} \quad (10)$$

With an energy term given by:

$$E(V, h) = - \sum_{i=1}^m \sum_{j=1}^F \sum_{k=1}^K W_{ij}^k h_j v_i^k - \sum_{i=1}^m \sum_{k=1}^K v_i^k b_i^k - \sum_{j=1}^F h_j b_j \quad (11)$$

The movies with missing ratings do not make any contribution to the energy function.

### 3.3 Learning

The parameter updates required to perform gradient ascent in the log-likelihood can be obtained from equation (10)  $p(V)$ :

$$\begin{aligned} \Delta W_{ij}^k &= \epsilon \frac{\partial \log p(V)}{\partial W_{ij}^k} = \\ &= \epsilon (\langle v_i^k h_j \rangle_{data} - \langle v_i^k h_j \rangle_{model}) \end{aligned} \quad (12)$$

Where  $\epsilon$  is the learning rate. The expectation  $\langle v_i^k h_j \rangle_{data}$  defines the frequency with which movie  $i$  with rating  $k$  and feature  $j$  are on together when the features are being driven by the observed user-rating data from the training set using equation (9)  $p(h_j = 1|h)$ , and  $\langle v_i^k h_j \rangle_{model}$  is an expectation with respect to the distribution defined by the model.

### 3.4 Making Recommendations

Restricted Boltzmann machine can model a rating distribution over the visible units. In order to infer the missing ratings, user ratings are clamped on the softmax units and single Gibbs sampling step is performed over all the missing ratings. The exact prediction algorithm is given below:

#### Algorithm: RBM – Making Recommendations

Input: a user  $u$ , an item  $i$

Output: an estimation of  $R(u, i)$

1. Clamp the ratings of  $u$  over the softmax units of the RBM.
2. Compute  $\hat{p}_j = p(h_j = 1|V)$  for all hidden units  $j$ .
3. Compute  $p(v_i^k = 1|\hat{p}) = \frac{\exp(b_i^k + \sum_{j=1}^F \hat{p}_j W_{ij}^k)}{\sum_{l=1}^K \exp(b_i^l + \sum_{j=1}^F \hat{p}_j W_{ij}^l)}$  for  $k = 1, \dots, K$ .
4. Take the expectation as prediction, i.e.,  $R(u, i) = \sum_{k=1}^K p(v_i^k = 1|\hat{p}) k$ .

### 3.5 Weighted Parallel Deep Hybrid Model

Hybrid weighting is a hybridization technique that computes the predication score of all recommendations approaches by giving weight to each approach and summing these weights to produce a new output recommendation. There are two ways to determine the weight, namely empirical bootstrapping and dynamic weighting. Both SVD and RBM recommendation approaches are combined using weighted strategy, the prediction score of users  $u$  to item  $i$  can be computed as follow

$$rec_{weighted}(u, i) = \sum_{k=1}^n \beta_k \times R_k(u, i) \quad (13)$$

Where  $\beta_k$  denotes the weight of algorithm  $R_k(u, i)$ . In proposed model, weighted hybridization combines two techniques, so set  $n = 2$ . Accordingly, if  $n = 2$ , the computation of predication score can be written as:

$$rec_{weighted}(u, i) = \beta_1 \cdot R_1(u, i) + (1 - \beta_1) \cdot R_2(u, i) \quad (14)$$

and the optimized weight can be gained by computing:

$$\beta_1 = \frac{\sum_u \sum_i (rec_{weighted}(u, i) - R_2(u, i)) (R_1(u, i) - R_2(u, i))}{\sum_u \sum_i (R_1(u, i) - R_2(u, i))^2} \quad (15)$$

Finally, the proposed weighted parallel hybrid model provides accurate prediction for personalized recommendation. The cold-start problem is alleviated by weighted hybridization prediction score of two algorithms as combines the benefit of both SVD and RBM.

#### 4. Results and Discussion

**Proposed A Weighted Deep Parallel hybrid method:** The Weighted Parallel hybrid model, makes use of user-item rating matrix as input. A weighted hybridization technique combines the two recommendation systems by computing weighed sums of their prediction rating scores. Both SVD and RBM algorithms have their own predication rating score.

The evaluation metrics RMSE and MAE are used to measure the performance among the different models. The five-fold cross-validation approach has been used in each dataset. The RMSE and MAE values achieved by the proposed model and the other algorithms in comparison are given in Table 2 and Table 3 for Movielens 100K and Movielens 1M correspondingly. The tables clearly show that the proposed hybrid method gives better performance in terms of both RMSE and MAE compared to other algorithms for both the dataset.

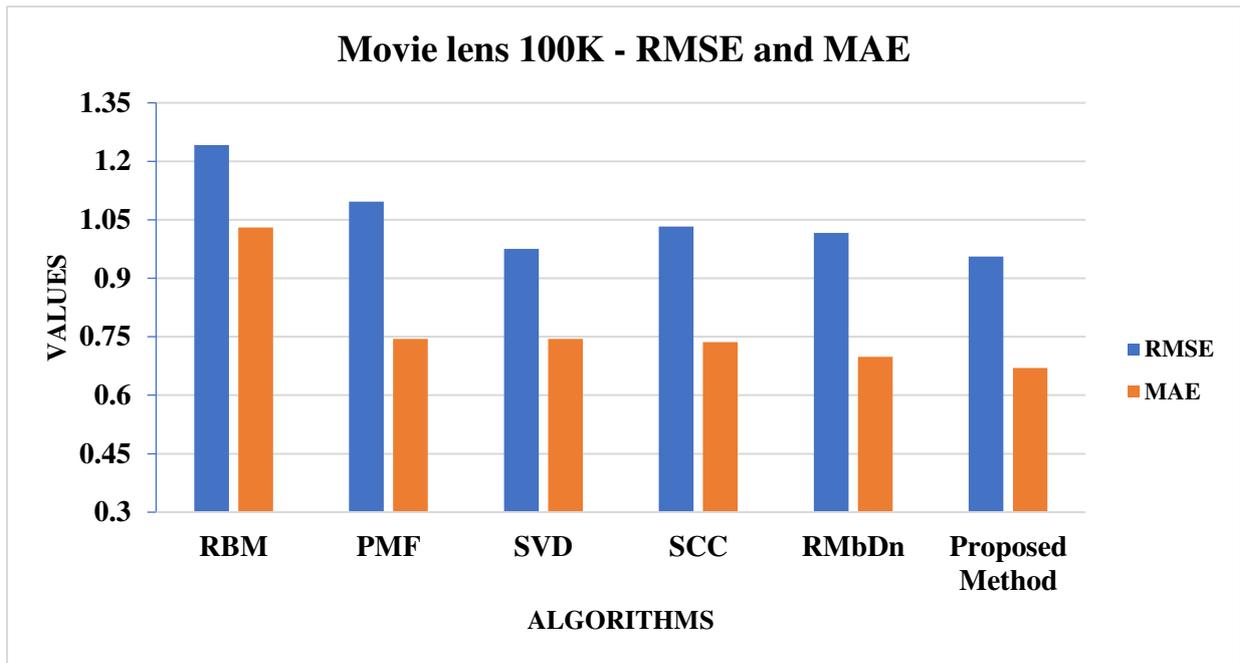
<b>Metric</b>	<b>RBM</b>	<b>PMF</b>	<b>SVD</b>	<b>SCC</b>	<b>RMbDn</b>	<b>Proposed Method</b>
<b>RMSE</b>	1.2418	1.0967	0.9753	1.0325	1.0165	<b>0.9557</b>
<b>MAE</b>	1.0303	0.7447	0.7446	0.7365	0.6985	<b>0.6699</b>

**Table 2.** The performance of RMSE and MAE on Movielens 100K.

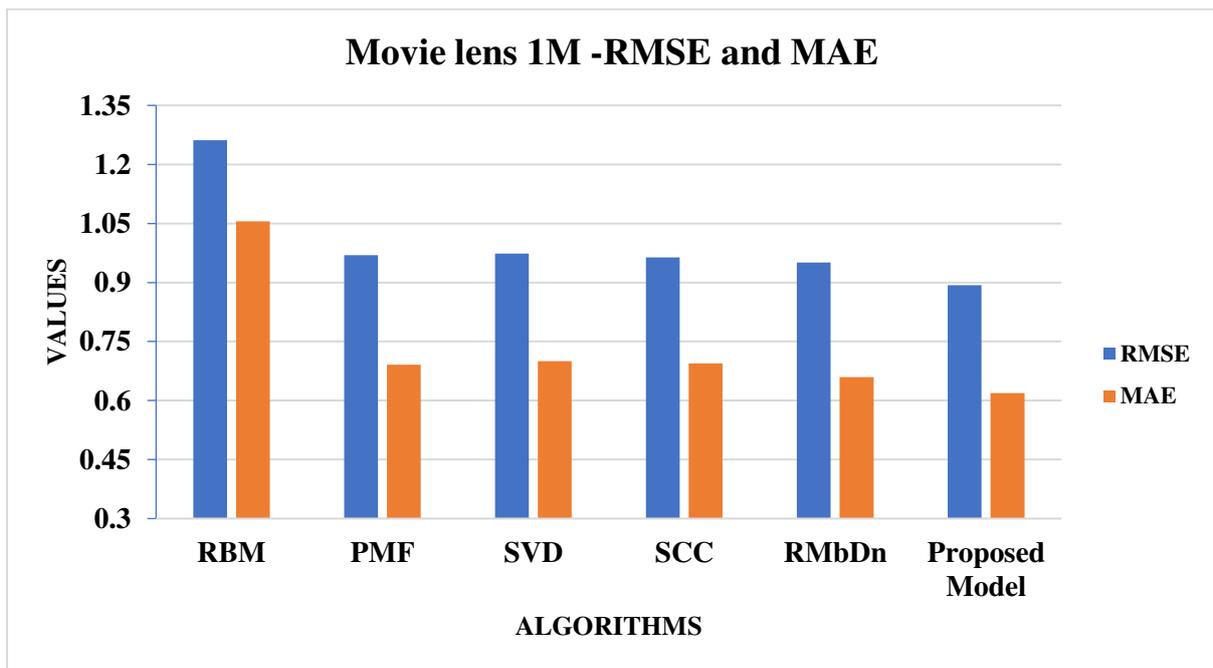
<b>Metric</b>	<b>RBM</b>	<b>PMF</b>	<b>SVD</b>	<b>SCC</b>	<b>RMbDn</b>	<b>Proposed Method</b>
<b>RMSE</b>	1.2615	0.9697	0.9736	0.9641	0.9508	<b>0.8931</b>
<b>MAE</b>	1.0558	0.6914	0.6997	0.6945	0.6598	<b>0.6187</b>

**Table 3.** The performance of RMSE and MAE on Movielens 1M.

The experimental results of the proposed model and corresponding comparison with state-of-the-art methods are presented in Figure 4 and 5. for two datasets. The value of RMSE and MAE of the proposed model on the MovieLens-100K datasets is 0.9557 and 0.6699. The value of RMSE and MAE of the proposed model on the MovieLens-1M datasets is 0.8931 and 0.6187. It is clear from the figures that the hybrid approach performs comparatively better than the individual methods namely, SVD and RBM.



**Figure. 4.** Performance of all methods on Movielens 100K.



**Figure. 5.** Performance of all methods on Movielens 1M.

Figure 6 and 7 shows the convergence of RMSE and MAE values for the proposed hybrid method and their corresponding individual algorithms namely, SVD and RBM using MovieLens-100K dataset correspondingly. The gradual decrease in both the metrics indicate the superiority of the hybrid approach

compared to their corresponding individual algorithms. The proposed algorithm performs relatively better than the others in comparison.

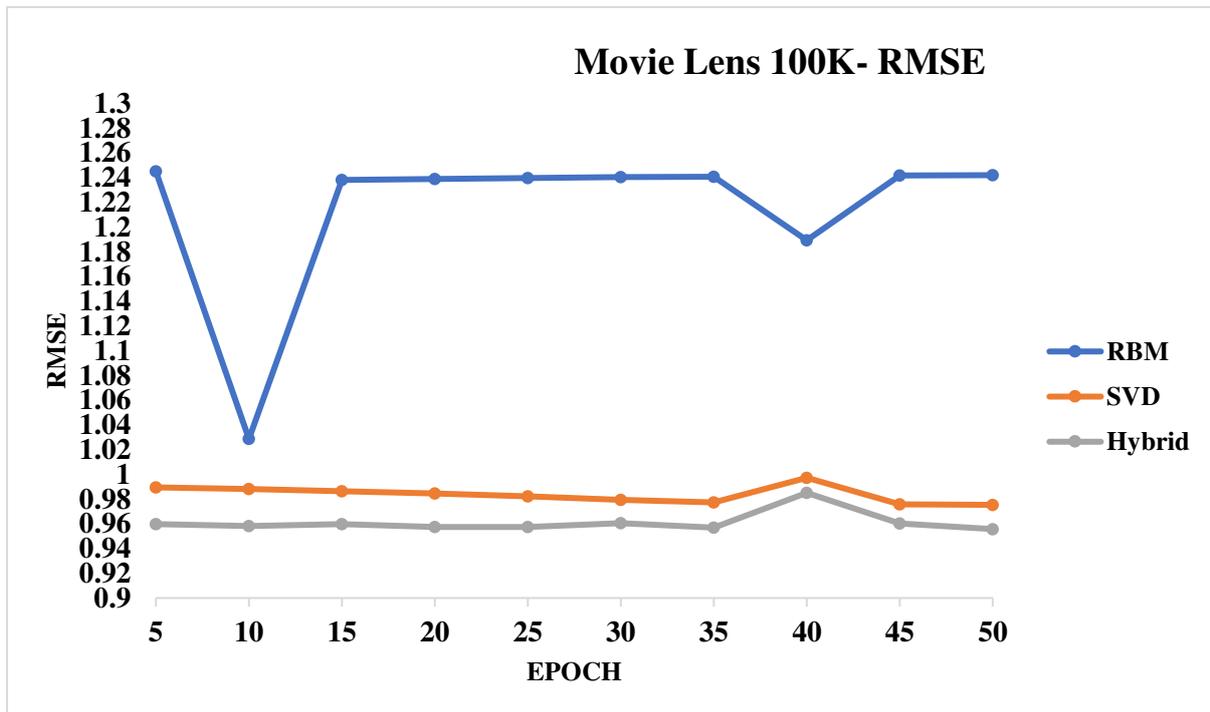


Figure. 6. RMSE performance on Movielens 100K.

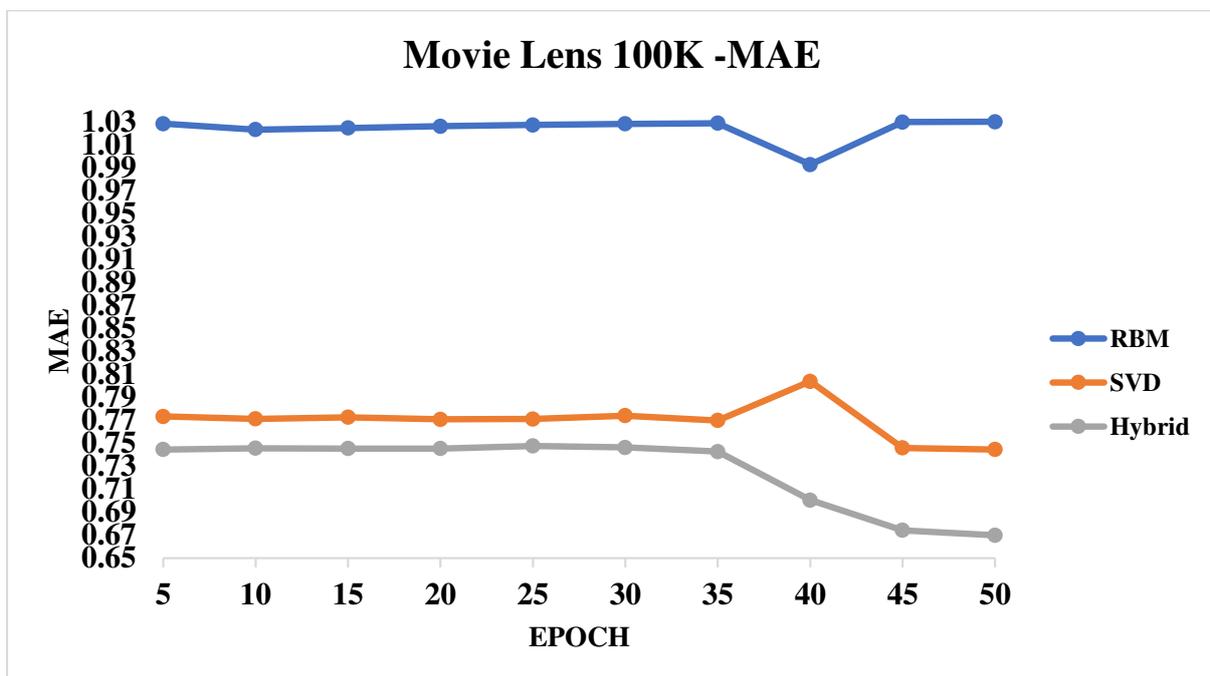


Figure. 7. MAE performance on Movielens 100K.

Based on the experimental results, the proposed hybrid model has the highest recommendation accuracy under both metrics on two datasets. The

proposed hybrid combination of the RBM and SVD achieves higher prediction and proves it more effective. This is attributed to the hybridization technique as it integrates the more robust SVD with RBM which aids in resolving the cold start problem. Thus, the hybridization of both these methods gives better performance. Hybrid recommendation technique provides most effective solution for building a recommender system. The main benefit of proposed Weighted Parallel Deep Hybrid Collaborative Technique which combines Singular Value Decomposition (SVD) and Restricted Boltzmann Machine (RBM) for recommendation is to suppress the drawbacks of traditional collaborative filtering technique or an individual technique in a combined model. The perfect combination of the two methods also aids to enhance the benefits of both the methods, where matrix factorization is achieved by SVD and learning to reduce cold start problem is achieved by RBM. Thus, the Weighted Hybridization increases the overall performance of the recommender system.

## **5. Methods**

### **5.1 Dataset**

In this section, to evaluate the performance of the proposed weighted parallel hybrid model two real MovieLens datasets: MovieLens - 100K and MovieLens - 1M is used. Both datasets contain user demographics. MovieLens is a movie rating dataset which was collected by GroupLens Research at the University of Minnesota. There are different kinds of this dataset available in <http://grouplens.org/datasets/movielens/>. The statistics of the two datasets are shown in Table 1.

<b>Dataset</b>	<b>Number of items</b>	<b>Number of users</b>	<b>Number of ratings</b>	<b>Number of ratings per user</b>	<b>Number of ratings per movies</b>	<b>Sparsity %</b>
MovieLens 100K	1682	943	100,000	106.41	59.451	93.70%
MovieLens of 1M	3952	6040	1,000,209	165.60	253.090	95.81%

**Table 1.** Description of the MovieLens Datasets

Both datasets also have additional information of users and item such as age, gender, occupation of users, release date and genres of movies.

## 5.2 Experimental setup

The proposed model is implemented on windows 10 and 64 – bit operating system running on Intel® Core™ i7-7500U CPU @ 2.70 GHz x64-based processor and hard disk of 500GB. This hybrid model is implemented in python language version 3.7 and tensor flow 3.0 as backend.

## 5.3 Parameter settings

The parameters for RBM are, the hidden dimension layer is 50, rating value is 10, learning rate is 0.001 and batch size is 100.

## 5.4 Evaluation Metric:

The most common Collaborative Filtering evaluation measure for prediction accuracy is the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), defined as follows,

$$RMSE = \sqrt{\frac{1}{N} \sum_{i,j} (R_{ij} - \hat{R}_{ij})^2} \quad (16)$$

$$MAE = \frac{1}{N} \sum_{i,j} |R_{ij} - \hat{R}_{ij}| \quad (17)$$

Where  $N$  is the number of testing data samples and  $R_{ij}$  is the rating of user  $i$  on the item  $j$  and  $\hat{R}_{ij}$  represents the predicted rating. A smaller value of RMSE and MAE indicates better performance of the method.

### 5.5 Performance analysis

The proposed hybrid algorithm is compared with the traditional recommendation algorithms given below.

**RBM (Restricted Boltzmann Machines)** <sup>16</sup>: RBM is a two-layer stochastic neural network consisting of visible and hidden units. To make the learning easier, restrict the network so that no visible unit is connected to any other visible unit and no hidden unit is connected to any other hidden unit.

**PMF (Probabilistic Matrix Factorization)** <sup>12</sup>: A widely used matrix factorization model. PMF is a model to factorize the user – item matrix in recommendation.

**SVD (Singular value decomposition)** <sup>14</sup>: SVD is the powerful technique of dimensionality reduction. In this algorithm, where the rating matrix is decomposed into three matrixes. These matrixes will be used to make prediction.

**SCC (Self- Constructing Clustering)** <sup>11</sup>: SCC is a recommendation algorithm based on clustering. It is an item-based clusters using a self-constructing clustering method.

**RMbDn (A recommendation model based on deep neural networks)** <sup>13</sup>: This recommendation based on QPR (Quadric Polynomial Regression) model to

obtain the latent features of user - item and combine with DNN (Deep Neural Network) model for predicting user's rating score.

## 6. Conclusion

The proposed hybrid collaborative filtering model, combines matrix factorization method SVD and deep learning technique RBM. The hybridization model can make use of both user-item rating matrix and side information for better prediction in recommendation. The performance of proposed model is evaluated through two different datasets namely, MovieLens - 100K and MovieLens - 1M using RMSE and MAE metrics. The five-fold cross-validation approach is used for evaluation. The experimental results show that the proposed hybridization model achieves better performance among the existing methods on both the datasets based on RMSE and MAE values.

## References

1. Atisha Sachan and Vineet Richhariya. Reduction of Data Sparsity in collaborative Filtering based on Fuzzy Inference Rules. *International Journal of Advanced Computer Research* 2013. **3**, Number-2, Issue-10,101-107.
2. Bingbing Dong, Yi Zhu, Lei Li and Xindong Wu. Hybrid Collaborative Recommendation via Dual-Autoencoder. *IEEE access* 2020, **8**, 46030 - 46040.
3. Dheeraj kumar Bokde, Sheetal Girase, Debajyoti Mukhopadhyay. Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey. *Elsevier* 2015, 136 -146.
4. Hong-Jian Xue, Xin-Yu Dai, Jianbing Zhang, Shujian Huang, Jiajun Chen. Deep Matrix Factorization Models for Recommender Systems. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)* 2017, 3203 – 3209.
5. Hinton, Geoffrey, E. Training products of experts by minimizing contrastive divergence. *Neural Comput.*2002,**14(8)**, 1771–1800.

6. Jiang Zhang, Yufeng Wang, Zhiyuan Yuan and Qun Jin. Personalized Real-Time Movie Recommendation System: Practical Prototype and Evaluation. *IEEE Tsinghua Science and Technology* 2019, **25**, Issue-2,180 – 191.
7. Kim, D., Park, C., Oh, J., Lee, S., and Yu, H. Convolutional matrix factorization for document context-aware recommendation. *in Proc. 10<sup>th</sup> ACM Conf. Recommender Syst. (RecSys)*, New York, NY, USA 2016, 233-240.
8. Kangkang Li, Xiuze Zhou, Fan Lin, Wenhua Zeng, and Gil Alterovitz. Deep Probabilistic Matrix Factorization Framework for Online Collaborative Filtering. *IEEE. Translations and content mining* 2019, **7**, 56117-56128.
9. Li, G., Hoi, S. C. H., Chang. K., Liu, W., and Jain, R. Collaborative online multitask learning. *IEEE Trans. Knowl. Data Eng* 2014, **26**,1866-1876.
10. Li et al, Sheng Li, Jaya Kawale, and Yun Fu. Deep collaborative filtering via marginalized denoising autoencoder. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management 2015*, 811-820.
11. Liao, C. L., and Lee, S. J. A clustering-based approach to improving the efficiency of collaborative filtering recommendation. *Electron. Commerce Res. Appl.*2016, **18**, 1- 9.
12. Liu, J., Wang, D., and Ding, Y. PHD: A probabilistic model of hybrid deep collaborative filtering for recommender systems. *in Proc. 9th Asian Conf. Mach. Learn. 2017*, **77**, 224 – 239.
13. Libo Zhang, Tiejian Luo, Fei Zhang, and Yanjun Wu. A Recommendation Model Based on Deep Neural Network. *IEEE Access* 2018, **6**, 9454 - 9463.
14. Mehrbakhsh Nilashi, Othman Ibrahima, Karamollah Bagherifardc. A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications* 2018, **92**, 507 – 520.
15. Ruihui Mu. A Survey of Recommender Systems Based on Deep Learning. *IEEE access* 2018, **6**, 69009 – 69022.
16. Salakhutdinov, R., Mnih, A., and Hinton, G. Restricted Boltzmann machines for collaborative filtering. *in Proc. 24th Int. Conf. Machine Learning, Corvallis, OR, USA 2007*, 791 - 798.

17. Shuai Zhang, Lina Yao, Aixin Sun, Yi Tay. Deep Learning based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys* 2018, **1**.
18. Van den Oord, A., Dieleman, S., and Schrauwen, B. Deep content-based music recommendation. *in Proc. 26th Int. Conf. Neural Inf. Process. Syst. (NIPS), Lake Tahoe, NV, USA 2013*, **2**, 2643 - 2651.
19. Xin Dong, Lei Yu, ZhonghuoWu, Yuxia Sun, Lingfeng Yuan, Fangxi Zhang. A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) 2017*, 1309 - 1315.
20. Yang, B., Lei, Y., Liu, J., and Li, W. Social collaborative by trust. *IEEE Trans. Pattern Anal. Mach. Intell.* 2016, **39**, 1633 - 1647.
21. Yu Liu, Shuai Wang, Shahrukh Khan, M., and Jieyu He. A Novel Deep Hybrid Recommender System Based on Auto-encoder with Neural Collaborative Filtering. *IEEE Big Data Mining and Analytics 2018*, **1**, 211 – 221.

## **Author information**

### Affiliations

Department of Computer Science and Engineering, K.L.N. College of Engineering, Pottapalayam, Sivagangai District, Tamilnadu, India.

## **Acknowledgements**

Authors appreciate and thanks K.L.N. College of engineering for their support.

## **Author contributions**

Corresponding author contributed to the overall idea of this research and wrote the main manuscript text. All authors reviewed the manuscript.

## **Competing interests**

The authors declare no competing interests.

## **Funding**

The author(s) received no financial support for the research, authorship, and/or publication of this article.

## **Tables captions**

Table 1. Description of the MovieLens Datasets.

Table 2. The performance of RMSE and MAE on MovieLens 100K.

Table 3. The performance of RMSE and MAE on MovieLens 1M.

## **Figure captions**

Figure 1. Architecture diagram of Proposed Hybrid Recommendation System.  
(Attached in separate PDF document)

Figure 2. A Framework of Restricted Boltzmann Machine.

Figure 3. Restricted Boltzmann Machine for Collaborative Filtering.

Figure 4. Performance of all methods on Movielens 100K.

Figure 5. Performance of all methods on Movielens 1M.

Figure 6. RMSE performance on Movielens 100K.

Figure 7. MAE performance on Movielens 100K.