Research Article

International Journal of Advanced Technology and Engineering Exploration, Vol 10(104)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2022.10100088

Evolutionary extreme learning machine based collaborative filtering

Pratibha Yadav[1], Shweta Tyagi[1*] and Harmeet Kaur[2]
Assistant Professor, Department of Computer Science, Shyama Prasad Mukherji College, University of Delhi, Delhi-110026, India[1]
Associate Professor, Department of Computer Science, Hansraj College, University of Delhi, Delhi-110007, India[3]

Received: 02-September-2022; Revised: 21-June-2023; Accepted: 23-June-2023

Abstract
Research in the field of collaborative filtering (CF) has demonstrated its importance and effectiveness compared to other recommendation engines such as content-based and hybrid recommendation systems. However, there is ongoing research in the field of CF to improve the list of recommended items and generate accurate recommendations when dealing with sparse datasets. To enhance the prediction quality of recommender systems, a method based on an evolutionary extreme learning machine (EELM) for CF has been proposed to address the issue of sparsity. First, the dataset is pre-processed by filling in missing rating values in the rating database. Additionally, the dataset is trimmed by removing items with a limited number of ratings. This tailored rating database is then utilized with the extreme learning machine(ELM) technique to improve the quality of recommendations in the domain of CF. Furthermore, to enhance the accuracy of the proposed recommendation method, an evolutionary genetic algorithm(GA) was employed to train the parameters of the ELM-based model. The variants of the proposed scheme are compared to traditional recommendation methods by computing metrics such as mean absolute error (MAE), root mean square error (RMSE), precision, recall, and F-measure. Empirical analysis consistently indicates that the proposed approach outperforms the traditional CF-based recommendation method. The average accuracy metrics MAE and RMSE are reduced by factors of 9.69% and 32.83%, respectively, using the proposed technique. Additionally, the proposed scheme improves the classification accuracy of the recommender system by an average of 11.54%.

Keywords
Recommender system, Collaborative filtering, Missing data prediction, Extreme learning machine, Genetic algorithm.

1.Introduction
Since their emergence in the middle of the 1990s, recommender systems (RSs) have drawn more interest from academics and businesses [1, 2]. These systems investigate and assess user data from social media platforms, including followers, followed, comments, likes, dislikes, tweets, posts, and more [3, 4]. Additionally, the Internet of Things (IoT) has developed into a new source of information that includes global positioning system (GPS) coordinates, radio frequency identification (RFID) data, health indicators [5] surveillance data, etc.

To provide consumers, with the suggestions or recommendations they want, RS uses a variety of information sources, opening a popular field for additional study and improvement [3, 4].

*Author for correspondence

The practical applications of RSs help users to find movies, news, music, books, jokes, digital products, applications, websites, travel destinations, and e-learning materials [6–9]. However, there are still many issues with recommendation systems that need to be resolved, including accuracy, speed, novelty, dispersity, stability, and privacy [4, 10, 11].

In literature, different techniques have been designed and developed to handle these issues and to improve the accuracy of predictions and recommendations to the target user. Collaborative filtering (CF) [12], Content-based filtering [13], and hybrid [14, 15] approaches are the common and well-researched recommendation methods. The CF technique creates recommendations for the active user by examining their affinity group's tastes [1]. In contrast, the content-based RS investigates the content of the documents that the active user has explored in the past to understand a user's taste and subsequently recommends additional items of interest [16].

Further, the hybrid RSs are designed to combine the different recommendation approaches to deal with the associated problems and to enhance the accuracy of recommendations [5, 17].

CF, one of the most well-known and commonly accepted recommendation systems, is broadly classified into three major categories namely memory-based, model-based, and hybrid methods [18, 19]. The memory-based CF recommendation techniques employ different measures such as Pearson correlation, cosine, and Jaccard similarity to select like-minded users for the target user [20]. After that, predictions are generated for the target user using the rating patterns of similar users [21, 22]. Model-based CF techniques are constructed over the history of users' behaviors. A learned user behavior model is then employed to anticipate the active user's behavior [23]. Hybrid CF recommendation systems combine CF approaches with additional recommendation techniques to analyse and predict user behaviour [23, 24].

CF techniques performance is adversely affected by the issues such as sparsity, scalability, and cold-start [17, 25]. Numerous approaches have been developed in the literature to tackle these issues. Regression-based recommendation algorithms have caught the attention of researchers and become one of the most common and effective approaches for tackling these issues. Regression-based recommendation algorithms have used various techniques such as linear regression [26], nonlinear regression model [27], polynomial regression [28], temporal regression [29], additive regression [30], logistic regression [31], ordinal regression [23] and pairwise preference regression [32]. Linear regression performs poorly when there is a non-linear relationship in the dataset. On the other hand, non-linear regression algorithms are usually very complex and time-consuming. Extreme learning machine (ELM) [33], one of the most frequently used regression algorithms, is known for its advantages of high accuracy and speedy learning. Since ELM is a single-hidden layer feed-forward neural network (SLFN), the layers do not need to be optimised to achieve accuracy. The selection of parameters, however, affects how well ELM-based RS performs. Our work focuses on optimising the parameters of ELM and thereby enhancing the performance of the RS. The main contributions of this study are as follows:

- Adopted the missing data prediction algorithm and further trimmed the dataset to address the problem of sparsity
- Additionally, ELM based regression model is suggested to provide improved recommendations to the active users
- An evolutionary genetic algorithm (GA) is employed to further enhance the efficiency of ELM by learning the associated parameters
- The ELM-based model uses several hidden layers, which impacts the quality of the model. The value of this parameter is selected empirically to improve the performance of the proposed model.
- To demonstrate the superiority of the suggested work above conventional recommendation methodologies, extensive empirical evaluation has been conducted

The remainder of this article is structured as follows. Section 2 briefly summarizes theories about CF-based RSs, ELM, and evolutionary GA. In section 3, a new model based on missing data prediction and Evolutionary ELM (EELM) for the CF-based recommendation method is proposed. In section 4, the proposed model with its variants is compared with traditional regression-based recommendation methods, and extensive experimental analysis is presented. Section 5 presents a discussion of the experiments conducted based on the proposed model and the limitations of the proposed work. Section 6, concludes the proposed model and highlights future work.

2.Literature review

CF is the most popular recommendation technique that suggests products to the active user based on the preferences of the affinity group. However, the prediction accuracy of the CF-based RS is adversely affected by key issues such as sparsity, scalability, and cold-start [4, 34]. To handle these problems different regression algorithms have been adopted in the domain of CF. The regression-based CF models generate more accurate recommendations for the active user. Frank and Hall suggested an additive regression model which is employed for a large-scale CF problem [30]. Mild and Natter developed CF or regression-based models for internet recommendation methods to provide significantly better recommendations than traditional CF methods [35]. Further, a CF-based recommendation scheme using a regression-based approach to efficiently address the problem of data sparsity and prediction latency was designed and implemented by Vucetic and Obradovic [36]. Chuan et al. developed a novel recommendation method incorporating item-based filtering and user-based classified regression providing personalized product recommendations [37]. Moreover, Zhu et al.

developed a CF-based RS by employing the polynomial regression method for providing secured recommendations [28]. Additionally, a collaborative tag recommendation system using a logistic regression-based system was designed by Montanés et al. [31]. Moreover, for handling the issue of cold start recommendations, Park and Chu proposed a pairwise preference regression model [32]. Further, to generate time-dependent collaborative personalized recommendations, Brenner et al. employed the temporal regression technique [29]. Purushotham et al. utilized the users' social information and items' content information to develop a collaborative topic regression model for providing quality recommendations to the user [38]. Further, Chang et al. proposed an ordinal regression model which employs singular value decomposition and support vector ordinal regression to generate recommendations under data of mild data sparsity and large-scale conditions [23]. A real-time CF (RCF) algorithm was proposed by Deng et al. [39] that adopts a regression-based ELM algorithm in the domain of CF to generate real-time recommendations. The ELM algorithm is the most effective and fast-working regression algorithm. The proposed work is motivated by the RCF algorithm and adopts ELM to generate more accurate recommendations. Further, evolutionary GA is adopted to optimize the parameters of ELM and to boost the performance of the model. The details of the working architecture of ELM and GA are explained next.

The ELM is an emerging and useful learning algorithm proposed by Huang et al. that provides efficient and effective output for generalized feed-forward networks [33]. Numerous networks have been suggested in the literature to study the learning behaviour and output produced by these models [40–44]. It initially originated from SLFN [33]. ELM has a substantial advantage over SLFN due to its simple installation and quick learning. The functioning of ELM is illustrated in *Figure 1*.

We can now understand the ELM algorithm for SLFNs with $\tilde{N}$ hidden neurons, activation function g and p training samples, $\{(x_i, t_i) \mid i = 1,2,\dots,p\}$ where $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d}) \in \mathbb{R}^d$, $t_i = (t_{i,1}, t_{i,2}, \dots, t_{i,d'}) \in \mathbb{R}^{d'}$, and $\tilde{N} \ll p$. This can be written as shown in Equation 1:

$$\sum_{j=1}^{\tilde{N}} \beta_i g(w_j . x_i + b_j) = o_i,\ i = 1,2,\dots,p, \quad (1)$$

Here $\beta_j = [\beta_{j,1}, \beta_{j,2}, \dots, \beta_{j,d'}]^T$ is the output weight vector linking the j-th hidden node and output nodes, $w_j = [w_{j,1}, w_{j,2}, \dots, w_{j,d}]^T$ is the input weight vector linking the j-th hidden neuron and the input neurons, and the b_j is the threshold of the j-th hidden node.

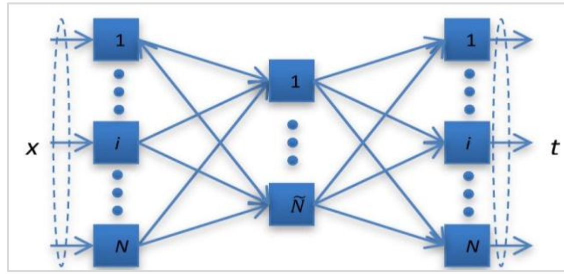

Figure 1 Functioning of ELM

The efficient arrangement of the p equations given above can be represented as shown in Equation 2:

$$H\beta = O \quad (2)$$

where Equation 3 and Equation 4 are,

$$H(w_1, \dots, w_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}, x_1, \dots, x_p) = \begin{bmatrix} g(w_1 . x_1 + b_1) & \dots & g(w_{\tilde{N}} . x_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 . x_p + b_1) & \dots & g(w_{\tilde{N}} . x_p + b_{\tilde{N}}) \end{bmatrix}_{p \times \tilde{N}} \quad (3)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times d'} \text{ and } O = \begin{bmatrix} o_1^T \\ \vdots \\ o_p^T \end{bmatrix}_{p \times d'} \quad (4)$$

Here, H represents the hidden layer output matrix [33, 41]. The j-th column of the matrix represents the j-th hidden neuron's output vector regarding inputs $x_1, x_2, \cdots, x_p$. Moreover, the values of parameters β_j, w_j, and b_j are selected with zero error such that as shown in Equation 5 and Equation 6:

$$\|o_i - t_i\| = 0, \qquad i = 1,2,\dots,p, \quad (5)$$

i.e.

$$\sum_{j=1}^{\tilde{N}} \beta_j g(w_j . x_i + b_j) = t_i, i = 1,2,\dots,p. \quad (6)$$

According to [36], the solution of Equation 2 is given as shown in Equation 7.

$$\beta = H^{\dagger} T. \quad (7)$$

where $H^{\dagger}$ is the Moore-Penrose generalized inverse [40] of matrix H. So, the approximate solution matrix Y can be specified by (Equation 8)

$$Y = H\beta = HH^{\dagger}T. \quad (8)$$

The procedure of the ELM algorithm is described in Algorithm 1.

Algorithm 1: ELM Algorithm
Input: Training set $\mathcal{N} = \{(\mathbf{x_i}, \mathbf{t_i}) \mid \mathbf{x_i} \in \mathbb{R}^{\mathbf{d}}, \mathbf{t_i} \in \mathbb{R}^{\mathbf{d'}}, \ \mathbf{i} = \mathbf{1}, \dots, \mathbf{p}\}$, $\tilde{\mathbf{N}}$ **= number of hidden nodes and** $\mathbf{g(w_j, b_j, x_i)}$ **is the activation function for** $\mathbf{j} = \mathbf{1}, \dots, \tilde{\mathbf{N}}$ **and** $\mathbf{i} = \mathbf{1}, \dots, \mathbf{p}$.
Output: Trained cases of ELM.
1) **Arbitrarily produce parameters of hidden node** $(\mathbf{w_j}, \mathbf{b_j})$, $\mathbf{w_j} \in \mathbb{R}^{\mathbf{d}}$, $\mathbf{b_j} \in \mathbb{R}$
2) **Compute hidden node output matrix H**
3) **Compute output weight vector** $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\,\mathbf{T}$

The randomly chosen parameters namely weight and bias heavily influence ELM performance. If the ELM's randomly generated parameters are left unchanged during the training phase, they could negatively impact the RS's accuracy. Researchers have proposed a number of ELM variations, including incremental ELM[45], pruning ELM [46], error-minimized ELM [47], two-stage ELM [48], online sequential ELM [49], EELM [50], voting-based ELM [51], ordinal ELM [52], fully complex ELM [53], and symmetric ELM [54]. Additionally, research in the field of ELM-based models discovered that GA can be used to enhance the ELM-based model's performance. GA is an evolutionary algorithm that is used to optimize the solution of a complex search problem [55]. The stages of the algorithm are shown in *Figure 2*.

GA starts searching for the optimal solution to the problem by operating reproduction, crossover, and mutation procedures on generations to produce a new generation and checks the fitness of each generation. The algorithm stops working when the stopping criterion is satisfied. Hazir et al. applied the GA to optimize the parameters of the support vector machine (SVM) and ELM to predict the adhesion strength with improved accuracy [56]. Further, Alencar et al. developed a new model based on a GA to prune hidden layer neurons and overcome the drawback of poor generalization of ELM [46].

In the field of CF, regression-based techniques employ either linear or non-linear regression algorithms. According to the literature review, linear regression techniques are simple yet effective, but the presence of non-linear correlations in the data causes the performance of linear regression-based models to suffer. There have been many non-linear regression techniques proposed to address the non-linearity problem. However, one of the primary problems that worries the researchers the most is the complexity of the non-linear regression. ELM, which can handle complex data through high-speed learning and produce more accurate results, was presented as a solution to this problem [33].

The suggested work applied ELM in the domain of CF to create a novel recommendation engine that enhanced the quality of recommendations by utilising the benefits and efficacy of the ELM algorithm.

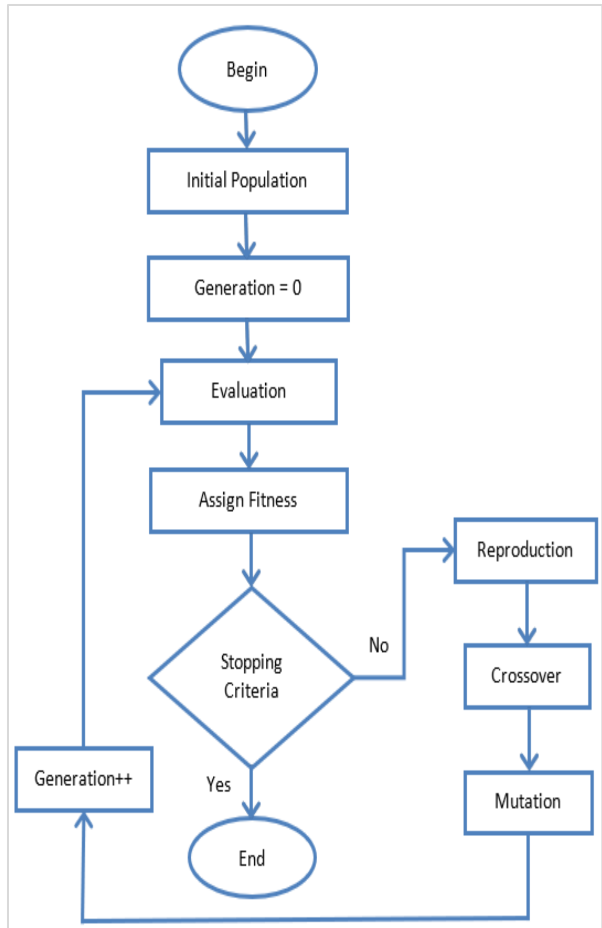

Figure 2 Genetic algorithm(GA)

3.Methods

Evolutionary extreme learning machine (EELM) based CF model

In this work, the complete task of recommendation is separated into offline and online processing steps. During offline processing, the missing rating values are estimated using an effective missing data prediction (EMDP) approach [57] to reduce the sparsity in the training dataset and then items dropped from this dataset, that are rated or seen hardly by the user. Thereafter, ELM-based and EELM-based models are employed separately during online processing in the domain of CF to generate recommendations for the active user. Moreover, these models further incorporate user-based, Item-based and, hybrid approaches. Consequently, we developed six variants of the proposed model; user-based ELM model for CF (U_ECF), Item-based ELM model for CF (I_ECF), Hybrid of U_ECF and I_ECF (H_ECF), User-based EELM model for CF (U_EECF), Item-based EELM model for CF (I_EECF), and Hybrid of U_EECF and I_EECF (H_EECF). The architecture of the proposed recommendation system model is depicted in *Figure 3.*

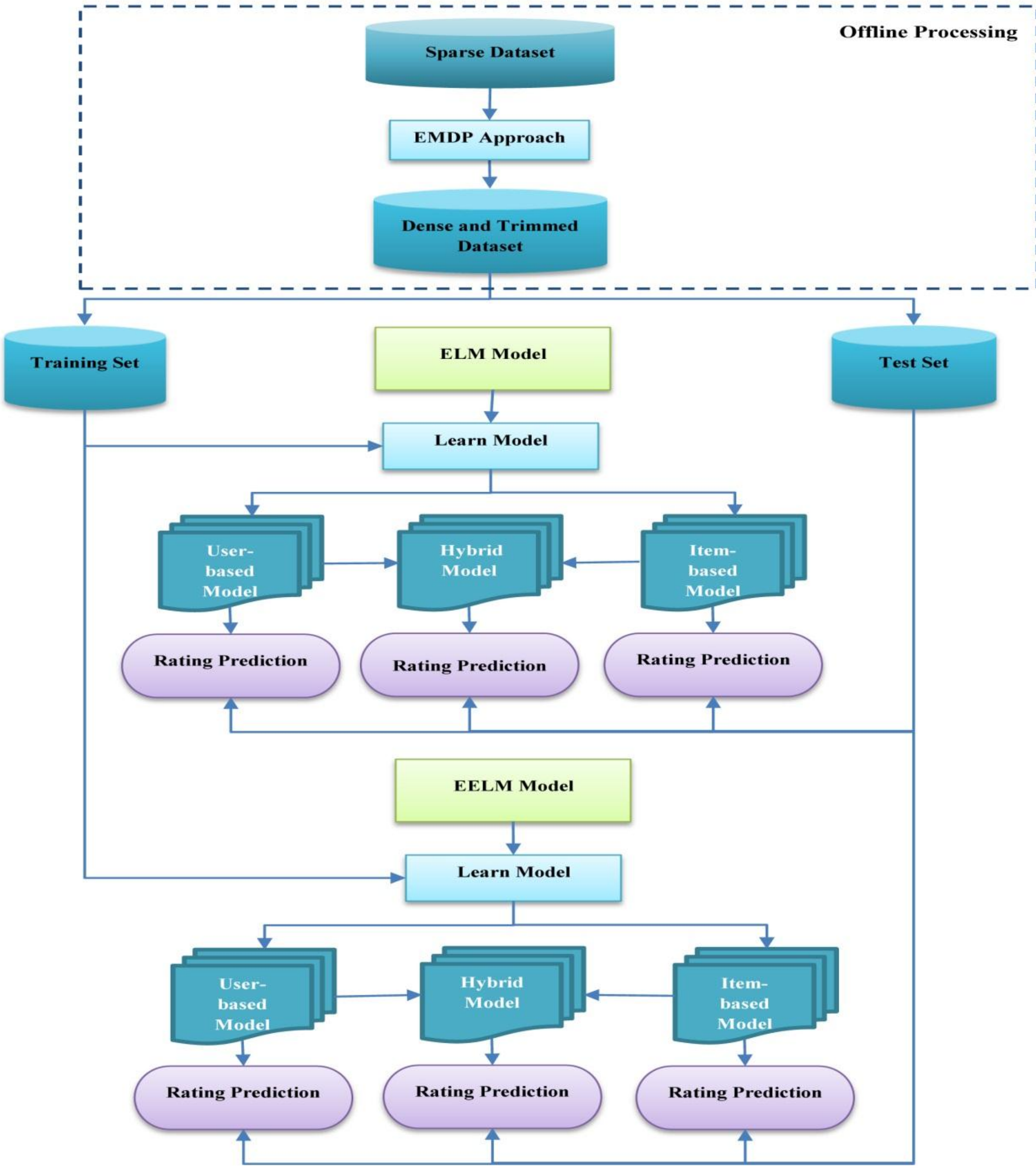

Figure 3 Architecture of proposed model

This work is inspired by the RCF model suggested by Deng et al. [39]. The RCF scheme is developed for improving the accuracy of CF by applying ELM on the rating dataset. This model converts the users' ratings in terms of likes and dislikes and then removes those items that escalate the sparsity in the dataset. In contrast, the proposed model tries to handle the problem of sparsity by first filling in missing values and removing the items with few ratings. Further, to enhance the accuracy of predictions, the proposed model considers the actual ratings of items to train the model rather than converting the rating values into likes and dislikes, unlike the RCF model. The phases of the proposed recommendation algorithm are elaborated in the subsequent subsections.

3.1Missing data prediction
CF-based RS makes the recommendation of items for the active user by exploring the interest of similar users. The similarity between users is measured by analyzing their pattern of ratings. Additionally, missing values in rating vectors affect the computation of similarity and thus penalize the accuracy of prediction. So, the proposed model fills the missing values present in the dataset during offline processing to tackle the issue of sparsity. For the prediction of missing rating values, the EMDP suggested by Ma et al. [58] is employed. The steps of the algorithm for EMDP are given below.

Algorithm 2: EMDP Algorithm

Input: Sparse Rating Matrix $R_S = \begin{bmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{N1} & \cdots & r_{Nm} \end{bmatrix}$

Output: Dense Rating Matrix $R_D = \begin{bmatrix} r'_{11} & \cdots & r'_{1m} \\ \vdots & \ddots & \vdots \\ r'_{N1} & \cdots & r'_{Nm} \end{bmatrix}$

For each user $\mathbf{u_i} = [\mathbf{r_{i1}}, \mathbf{r_{i2}}, \dots, \mathbf{r_{im}}]$

$\quad \mathbf{U_i} \leftarrow \{\mathbf{u} \in \mathbf{U}: \mathbf{sim}(\mathbf{u_i}, \mathbf{u}) > \boldsymbol{\eta},\ \mathbf{u_i} \neq \mathbf{u}\}$

End

For each item $\mathbf{z_j} = [\mathbf{r_{1j}}, \mathbf{r_{2j}}, \dots, \mathbf{r_{Nj}}]$

$\quad \mathbf{z_j} \leftarrow \{\mathbf{z} \in \mathbf{Z}: \mathbf{sim}(\mathbf{z_j}, \mathbf{z}) > \boldsymbol{\theta},\ \mathbf{z_j} \neq \mathbf{z}\}$

End

For $\mathbf{i^{th}}$ **user** $\mathbf{u_i}$ **and** $\mathbf{j^{th}}$ **item** $\mathbf{z_j}$

$\mathbf{Ur'_{i,j}} \leftarrow$ **User-based Prediction**

$\mathbf{Zr'_{i,j}} \leftarrow$ **Item-based Prediction**

$\quad \mathbf{r'_{i,j}} \leftarrow \boldsymbol{\lambda} \times \mathbf{Ur'_{i,j}} + (\mathbf{1} - \boldsymbol{\lambda}) \times \mathbf{Zr'_{i,j}}$

End

The EMDP algorithm considers the parameters η, θ and λ which are selected experimentally as explained by Ma et al. [58].

3.2Trimming of dataset
After filling in the missing values, the dense rating matrix is processed again to discard infrequent items. For this purpose, the proposed Trimming Algorithm searches the entire dense rating dataset and retrieves those items which are rated by at least k users. The value of k is selected empirically.

Algorithm 3: Trimming Algorithm

Input: Dense Rating Matrix $R_D = \begin{bmatrix} r'_{11} & \cdots & r'_{1m} \\ \vdots & \ddots & \vdots \\ r'_{N1} & \cdots & r'_{Nm} \end{bmatrix}$

Output: Trimmed Rating Matrix $R_T = \begin{bmatrix} r''_{11} & \cdots & r''_{1M} \\ \vdots & \ddots & \vdots \\ r''_{N1} & \cdots & r''_{NM} \end{bmatrix}$

$\boldsymbol{R_T} \leftarrow [\ ]$

For $\mathbf{j} = \mathbf{1}: \mathbf{m}$

$\quad \boldsymbol{Z_j} \leftarrow [\boldsymbol{r'_{1j}}, \boldsymbol{r'_{2j}}, \dots, \boldsymbol{r'_{Nj}}]'$

$\quad \boldsymbol{\bar{Z}_j} \leftarrow \{\boldsymbol{r'_{ij}} \in \boldsymbol{Z_j}: \boldsymbol{r'_{ij}} \neq \mathbf{0}\}$

$\quad$ **If** $|\boldsymbol{\bar{Z}_j}| > \boldsymbol{k}$

$\quad\quad \boldsymbol{R_T} \leftarrow [\boldsymbol{R_T}: \boldsymbol{Z_j}]$

$\quad$ **End**

End

Using this algorithm, we can discard the items that increase the sparsity in the dataset and do not contribute useful information but increase the unnecessary computation.

3.3EELM based CF
The proposed ELM/EELM-based recommendation scheme is applied to the processed dataset and consists of three steps: (1) ELM/EELM-based Training scheme, (2) Prediction of ratings and, (3) Recommendation of items for an active user.

3.3.1EELM based training scheme
The trimmed rating set R_T, generated from the procedure explained previously, is considered for the application of the proposed recommendation scheme. For R_T, let $U = \{u_1, u_2, \dots, u_N\}$ be the set of users and $Z = \{z_1, z_2, \dots, z_M\}$ be the set of items. The filtered rating matrix R_T is given in *Table 1* where the i, j^{th} element, r''_{ij}, represents the rating of j^{th} item given by i^{th} user. The matrix R_T is divided row-wise into two sub-matrices A_1 and A_2 and column-wise into two sub-matrices R_1 and R_2 as shown in *Table 1.*

The sub-matrices A_1 and A_2 are used as training set and test set respectively. Whereas sub-matrices R_1 and R_2 are used as input and output during the training process of the EELM model. In this way, the rating matrix is divided into four portions R_1A_1, R_2A_1, R_1A_2 and R_2A_2. During the training process of the User-based ELM/EELM model, the two sub-matrices R_1A_1 and R_2A_1 are adopted as input and output respectively as shown in *Figure 4 (a).* Similarly, the two sub-matrices R_1A_2 and R_2A_2 are employed as input and output respectively during the prediction process of the proposed model.

Table 1 Rating matrix: R_T

User↓ /Item→	i_1	i_2	...	i_p	i_{p+1}	i_{p+2}	...	i_M	
u_1	5	3	...	1	5	4	...	1	R_1
u_2	1	2	...	4	3	3	...	3	
⋮	⋮	⋮	...	⋮	⋮	⋮	...	⋮	
u_d	1	1	...	2	2	3	...	2	
u_{d+1}	4	2	...	1	?	?	?	?	R_2
⋮	⋮	⋮	...	⋮	?	?	?	?	
u_{N-1}	3	4	...	4	?	?	?	?	
u_N	2	5	...	3	?	?	?	?	
		A_1				A_2			

The column vectors of matrix $R_1\,A_1$, corresponding to ratings of p items given by d users, are input instances. Further, the column vectors of matrix $R_2\,A_1$ are considered as the output instances corresponding to p items and $N-d=d'$ users. Similarly, the column vectors of matrix $R_1\,A_2$, corresponding to $M-p=q$ items and d users, are input instances. Furthermore, the column vectors of matrix $R_2\,A_2$ are considered as the output instances corresponding to q items and $N-d=d'$ users. The discussed training scheme is employed for the user-based approach. Whereas, the item-based ELM/EELM approach takes the transpose of Rating Matrix R_T ($R_T{}'$) for learning the model as shown in *Figure 4 (b).*

The training set $A_1=\{(x_i,t_i)\mid i=1,\dots,p\}$ is considered, for understanding the procedure of ELM/EELM algorithm. Where, $x_i=\left[r''_{1,i},r''_{2,i},\dots,r''_{d,i}\right]=[\,R_1A_1]_{1:d,i}$ and $t_i=\left[r''_{d+1,i},r''_{d+2,i},\dots,r''_{N,i}\right]=[\,R_2A_1]_{d+1:N,i}$. The proposed approach is applied to the input-output instances (x,t) of set A_1 and the trained model is then employed on the input records R_1A_2 of the test set A_2 to predict the value of its output instances in the subset R_2A_2. Further, with $\widetilde{N}$ hidden nodes and the activation function $g\left(w_j,b_j,x\right)$, for $j=1,\dots,\widetilde{N}$ the training algorithm works as follows.

$$\begin{bmatrix}[\,R_1\,A_1]_{dXp} & [\,R_1\,A_2]_{dXq}\\ [\,R_2\,A_1]_{d'Xp} & [\,R_2\,A_2]_{d'Xq}\end{bmatrix}$$

(a)

$$\begin{bmatrix}[\,R_1\,A_1]'_{pXd} & [\,R_2\,A_1]'_{pXd'}\\ [\,R_1\,A_2]'_{qXd} & [\,R_2\,A_2]'_{qXd'}\end{bmatrix}$$

(b)

Figure 4 Division scheme of rating matrix R^T for (a) User-based approach and (b) Item-based approach

Algorithm 4: Training Algorithm
Input: Training Model, Trimmed Rating Matrix subset $\mathbf{A_1}$, the number of hidden nodes $\widetilde{\mathbf{N}}$ and activation function g
Output: A trained instance of ELM/EELM with β.

1. **If (Training Model = = ELM)**
   $(\mathbf{w_j},\mathbf{b_j}) \leftarrow \mathbf{Random\_Selection}(\,)$, $\mathbf{w_j}\in\mathbf{R^p}$, $\mathbf{b_j}\in\mathbf{R}$
   If (Training Model = = EELM)
   $(\mathbf{w_j},\mathbf{b_j}) \leftarrow \mathbf{Genetic\_Algo}(\,)$, $\mathbf{w_j}\in\mathbf{R^p}$, $\mathbf{b_j}\in\mathbf{R}$
2. **Compute hidden node output matrix H using Equation 3 for $\mathbf{R_1A_1}$**
3. **Compute the output matrix T using Equation 2**
4. **Repeat steps (1) to (3) until $\|\mathbf{R_2A_1}-\mathbf{T}\|=\mathbf{0}$**
5. **Compute output weight vector $\boldsymbol{\beta}=\mathbf{H^\dagger\,T}$.**

This algorithm for the given input and output rating matrix selects the values of parameters iteratively until the termination criterion is satisfied. In this way, the steps of the ***Training Algorithm*** are employed to learn the value of the output weight vector β. In step 1, the values of parameters w_j and b_j are set by learning through a GA given in the algorithm $Genetic\_Algo$ for the EELM-based model. Whereas, the values of these parameters are selected randomly for the ELM-based model. Further, the most popular Rectified Linear Unit (ReLU) function [59] is considered the activation function g, which is defined as the positive part of its argument (Equation 9).

$$g(y_i)=max(0,y_i), i=1,2,\dots,\widetilde{N} \tag{9}$$

The ReLU function was first introduced by Hahnloser et al. [57] and it is considered to be computationally more efficient than sigmoid. In step 2, the value of the activation function is employed for the computation of the hidden matrix H. Moreover, for computing the Moore-Penrose generalized inverse [44] in step 5, the following Equation 10 is used.

$$H^\dagger=(H^TH)^{-1}H^T \tag{10}$$

Here the matrix $H^T H$ is non-singular. Moreover, to learn the values of parameters w_j and b_j, the detailed $Genetic\_Algo$ is adopted and the steps of the algorithm are given below.

Algorithm 5: Genetic_Algo
Input: $\mathbf{A_1, Pop \leftarrow Initial_{Population}, k \leftarrow 0,}$
$\mathbf{BestFit \leftarrow 0}$
Output: $\mathbf{weight\ w_j \in R^d}$ **and bias** $\mathbf{b_j \in R}$ **for** $\mathbf{j = 1, \ldots, \tilde{N}}$
$\mathbf{For\ i = 1 : sizeof(Pop)}$
    $\mathbf{if\ Fitness\ of\ Pop_i > BestFit}$
        $\mathbf{BestFit \leftarrow Fitness\ of\ Pop_i}$
    End
End
$\mathbf{While\ (Fitness\ Threshold <= BestFit\ ||\ k < \#Generations)}$
$\mathbf{BestTwo \leftarrow SelectBestChrom(Pop)}$
$\mathbf{LeastTwo \leftarrow SelectLeastFitChrom(Pop)}$
$\mathbf{NewChrom \leftarrow Crossover\&Mutation(BestTwo)}$
$\mathbf{NewPop \leftarrow (Pop - LeastTwo) \cup NewChrom}$
$\mathbf{Pop \leftarrow NewPop}$
$\mathbf{k \leftarrow k + 1}$
    $\mathbf{For\ i = 1 : sizeof(Pop)}$
        $\mathbf{If\ Fitness\ of\ Pop_i > BestFit}$
            $\mathbf{BestFit \leftarrow Fitness\ of\ Pop_i}$
        End
    End
End

In this algorithm, the initial population is selected randomly, which contains a set of chromosomes. The structure of the chromosome is shown in *Figure 5*. Each chromosome is a binary matrix of size $\tilde{N} \times (p+1)$, in which the j^{th} row contains the first p components corresponding to the weight vector w_j and the last component corresponding to the bias vector b_j. The fitness of each chromosome is computed by measuring the value of the F-Measure for the predicted and actual ratings of items on the training dataset. The algorithm stops by finding the best chromosome till the #Generations or the Fitness is greater than the value of Fitness Threshold.

$$\begin{bmatrix} w_{1,1} w_{1,2} \ldots\ldots\ldots w_{1,p} b_1 \\ . \\ . \\ . \\ w_{\tilde{N},1} w_{\tilde{N},2} \ldots\ldots\ldots w_{\tilde{N},p} b_{\tilde{N}} \end{bmatrix}$$

Figure 5 Chromosome structure

The user-based and item-based variants of the proposed ELM/EELM algorithm are adopted on Rating Matrices R_T and $R_T{}'$, respectively.

3.3.2Prediction of items for an active user

After learning the model, the value of β is further adopted for the prediction of ratings of items for the active user. The following algorithm is used for the process of prediction for User-based ELM/EELM.

Algorithm 6: Prediction Algorithm
Input: Training Model, Active users' Rating Matrix $\boldsymbol{R_1A_2}$
Output: Active users' Predicted Rating Matrix $\boldsymbol{R_2A_2}$

1) **If (Training Model = = ELM)**
    $\boldsymbol{(w_j, b_j) \leftarrow Random\_Selection}$
   If (Training Model = = EELM)
    $\boldsymbol{(w_j, b_j) \leftarrow Genetic\_Algo}$
2) **Compute hidden node output matrix H using Equation 3**
3) **Compute the output matrix** $\boldsymbol{R_2A_2}$ **using Equation 2**

Using this algorithm, the predictions for the active user are stored in the matrix R_2A_2. The same prediction scheme is employed for both schemes namely, User-based and Item-based. The training and prediction algorithm are implemented separately for these schemes.

3.3.3Recommendation of Items

In literature, two strategies have been adopted to recommend the items to the active user after generating predictions of items [60]. According to the top-n method [61], items with a high score are recommended to the user whereas the other recommendation methods [62] select a value of the threshold. The items getting a score more than the threshold are then recommended to the user. In this work, we adopted both methods separately to generate a list of recommendations for the active user which are discussed in the next section.

The prediction of items is made by User-based and Item-based approaches of ELM/EELM model. For prediction of item i by user u, User-based prediction and Item-based predictions are represented by $U\_P(u,i)$ and $I\_P(u,i)$ respectively. Moreover, ratings are also predicted using the Hybrid approach as suggested by Yadav and Tyagi [24]. The prediction of the rating of item i by user u using a Hybrid scheme, $H\_P(u,i)$, is given by the following formula Equation 11.

$$H\_P(u,i) = \lambda \times I\_P(u,i) + (1-\lambda) \times U\_P(u,i) \tag{11}$$

The prediction formula is a weighted average of predictions made by User-based and Item-based approaches. The selection of parameter λ is given in the next Section.

4.Results

The empirical analysis is done to check the performance of the proposed scheme contrary to traditional CF-based recommendation schemes. MATLAB R2018a platform has been used for the implementation of variants of the proposed algorithms on 11th Gen Intel(R) Core (TM) with 16GB RAM. For the purpose of evaluation of the prediction and classification accuracy of the novel approach, various experiments were carried out and the details are given below.

4.1Dataset

The MovieLens dataset was used for experimentation. It is a well-known movies database that comprises 1 million ratings of 4000 movies from 6000 users on a rating scale of (1-5). In this dataset, a 0 value indicates that the item is not rated by the user. A numerical scale for ratings represents 1 as bad and 5 as excellent. In the dataset, each user has provided ratings for at least 20 movies and each movie has been rated by at least one user. The demographic detail of each user as age, gender, occupation, and zip code has been provided in the dataset. Basic information about each movie as genre and release date is also given in the data set.

4.2Metrices

To examine the performance of the recommendation approaches, various metrics are proposed in the literature [60]. For comparative analysis, the MAE (Mean Absolute Error), RMSE (Root Mean Square Error), precision, recall, and F-measure metrics are incorporated.

4.2.1MAE

It examines the prediction accuracy of the technique by calculating the difference between the predicted and actual ratings of an item for a user. It is defined as below (Equation 12):

$$MAE = \frac{\sum_{i=1}^{N} |pred_{u,i} - orig_{u,i}|}{r_{total}} \tag{12}$$

where, $pred_{u,i}$ and $orig_{u,i}$ represent the predicted and actual rating of item i for user u respectively and r_{total} refers to the total number of the ratings. We remark that the absolute difference between the predicted and actual value, $|pred_{u,i} - orig_{u,i}|$, informs about the error in the prediction.

4.2.2RMSE

It examines the accuracy of prediction by taking into consideration the standard deviation of prediction error. It is expressed as below (Equation 13):

$$RMSE = \sqrt{\left[\frac{\sum_{i=1}^{N} (pred_{u,i} - orig_{u,i})^2}{r_{total}}\right]} \tag{13}$$

where, $pred_{u,i}$ and $orig_{u,i}$ represents the predicted and original rating of item i by user u respectively.

4.2.3Precision

Precision calculates the ability to make relevant recommendations out of the total recommendations and is defined by the following formula.

4.2.4 Recall

Recall is described as the ability to make relevant recommendations out of the set of total significant items (Equation 14).

$$Recall = \frac{Significant\ recommendations}{Total\ significant\ items} \tag{14}$$

4.2.5 F-Measure

F-measure computes the classification accuracy of the scheme. It is defined as follows Equation 15:

$$F-measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{15}$$

This measure captures the harmonic mean of Precision and Recall and reveals the classification accuracy of the model. The high value of F-measure indicates the improved classification accuracy whereas low values indicate lower classification accuracy.

4.3Experimental design

For empirical analysis, four samples of 500 training users were selected randomly from the dataset. Furthermore, to tackle the issue of sparsity, the extracted training sample is organized by filling in zero values using missing data prediction. In addition, the resulting training sample is further organized by removing infrequent items from the sample. Infrequent items refer to those items that have not been rated by at least 20 users. Thus, as a result, a trimmed dataset is obtained for effective training of samples with a minimum level of sparsity. Moreover, for testing purposes, 50 random active users have been extracted from the remaining original dataset. Additionally, for each active user, 70% of items are categorized as a training set of items and, the remaining 30% as a test set of items.

The variants of the proposed scheme are compared against the state-of-the-art schemes namely EMDP [57] and RCF [39]. The terms used for variants of the proposed approach and their description is given in *Table 2*. EMDP technique predicts ratings for a user by adopting a missing data prediction approach for handling the issue of sparsity [58]. Whereas, the RCF model takes into account the concept of ELM for making more accurate recommendations for an active user [39].

Table 2 List of the techniques

Technique		**Description**
State-of-the-art	EMDP	Effective Missing Data Prediction [58]
	RCF	Real-time CF [39]
Proposed	U_ECF	User-based ELM model for CF
	I_ECF	Item-based ELM model for CF
	H_ECF	Hybrid of U_ECF and I_ECF
	U_EECF	User-based Evolutionary ELM model for CF
	I_EECF	Item-based Evolutionary ELM model for CF
	H_EECF	Hybrid of U_EECF and I_EECF

4.4Comparative analysis

The proposed approach takes into consideration the abilities of both techniques EMDP and RCF and further fine-tunes the efficiency of recommendations by adopting the approach of the GA. The accuracy of the proposed model and the impact of different parameters on the proposed approach are discussed in the following subsections.

4.4.1Prediction accuracy

To observe the prediction accuracy, the proposed schemes were analyzed against EMDP on the basis of MAE and RMSE measures as shown in *Table 3*. For the computation of prediction accuracy top-n approach [61], of recommendation is adopted. Moreover, the experiments were conducted on four samples of the dataset and the values of different parameters are selected empirically which are discussed in the next section.

The variants of the proposed scheme namely U_ECF, I_ECF, and H_ECF outperform the state-of-the-art EMDP scheme and the results are summarized in *Table 3*. In addition, it is also observed that the prediction accuracy of the proposed hybrid scheme H_ECF is better than the proposed U_ECF and I_ECF schemes consistently. Similarly, the prediction accuracy of the proposed hybrid scheme H_EECF is better than the proposed U_EECF and I_EECF schemes consistently. Further, it is also observed that the accuracy of U_EECF, I_EECF, and H_EECF schemes is better than U_ECF, I_ECF, and H_ECF schemes respectively. Moreover, the proposed H_EECF scheme outperforms the rest of the variants of the proposed scheme as given in *Table 3*.

Table 3 Comparison of I_ECF, U_ECF, H_ECF, U_EECF, I_EECF and H_EECF against EMDP based on MAE and RMSE

Metric	**Sample**	**Technique**						
		State-of-the-art	**Proposed**					
		EMDP	**U_ ECF**	**I_ECF**	**H_ ECF**	**U_ EECF**	**I_EECF**	**H_ EECF**
MAE	**Sample 1**	0.937214	0.874156	0.863421	0.850795	0.827107	0.810174	0.802553
	Sample 2	0.904531	0.892532	0.874052	0.847571	0.869393	0.854234	0.844712
	Sample 3	0.899204	0.861643	0.852289	0.837025	0.822535	0.817082	0.795413
	Sample 4	0.874491	0.858237	0.840037	0.821842	0.818017	0.815208	0.812658
RMSE	**Sample 1**	1.591187	1.420136	1.061195	1.074775	1.070845	1.044914	1.023877
	Sample 2	1.775443	1.424455	1.371520	1.256311	1.128676	1.099832	1.019175
	Sample 3	1.420837	1.343097	1.237463	1.219042	1.188631	1.176028	1.104869
	Sample 4	1.513574	1.431211	1.361864	1.224964	1.205274	1.196054	1.084311

4.4.2Classification accuracy

The classification accuracy of the variants of the proposed scheme and the RCF model is observed by computing Precision, Recall, and F-measure. For comparative analysis, the concept of real-time updating explained in the RCF model, is considered in the proposed model to generate nine cases as given by Deng et al. [39]. For all the cases, experiments were conducted to scrutinize the accuracy of recommendations made by the variants of the novel technique over the RCF technique based on classification accuracy metrics.

For the computation of classification accuracy measures, items with a predicted score more than the average rating value, are considered for recommendation. Further, the classification accuracy computed based on the precision, recall, and F-measure evaluation metrics are given in *Table 4*, *Table 5*, and *Table 6*, respectively. The experiments were conducted for the four samples separately, then the average was obtained. Experimental results reveal that the classification accuracy of the proposed schemes U_ECF, I_ECF, H_ECF, U_EECF, I_EECF, and H_EECF is better than the RCF scheme in all cases. Further, the proposed hybrid scheme H_ECF considerably improves the accuracy of proposed schemes U_ECF and I_ECF when compared based on Precision, Recall and, F-measure. Similarly, the proposed hybrid scheme H_EECF significantly improves the accuracy of proposed schemes U_EECF and I_EECF. Furthermore, H_EECF performs better than H_ECF and outperforms the other variants of the proposed scheme.

Table 4 Comparison of RCF and proposed techniques on the basis of precision

Case	RCF	U_ECF	I_ECF	H_ECF	U_EECF	I_EECF	H_EECF
Case A	0.778552	0.730841	0.717647	0.775764	0.800739	0.801493	0.810606
Case B	0.766071	0.727778	0.722428	0.764226	0.802381	0.806102	0.810606
Case C	0.780425	0.714444	0.729913	0.760788	0.817141	0.802643	0.816102
Case D	0.790374	0.722227	0.733684	0.761157	0.814004	0.810606	0.802643
Case E	0.782352	0.730694	0.719835	0.767895	0.810606	0.816102	0.802756
Case F	0.793899	0.731974	0.718583	0.761579	0.816102	0.802381	0.798276
Case G	0.784097	0.722341	0.734615	0.771721	0.802643	0.810606	0.819458
Case H	0.774659	0.730174	0.717925	0.761342	0.802756	0.816102	0.804008

Table 5 Comparison of RCF and proposed techniques on the basis of recall

Case	RCF	U_ECF	I_ECF	H_ECF	U_EECF	I_EECF	H_EECF
Case A	0.703782	0.741253	0.817213	0.807111	0.829189	0.815373	0.822119
Case B	0.808665	0.783806	0.842974	0.836456	0.82619	0.837453	0.829189
Case C	0.607946	0.761007	0.802719	0.840971	0.833193	0.822277	0.837453
Case D	0.687234	0.821897	0.823995	0.845485	0.81688	0.829189	0.822277
Case E	0.680141	0.809811	0.835948	0.834199	0.829189	0.837453	0.836226
Case F	0.69669	0.821631	0.840632	0.827427	0.830559	0.826192	0.823108
Case G	0.675413	0.810187	0.828723	0.780023	0.822277	0.829189	0.824724
Case H	0.658865	0.805083	0.819555	0.847743	0.836226	0.837453	0.839177

Table 6 Comparison of RCF and proposed techniques on the basis of F-Measure

Case	RCF	U_ECF	H_ECF	I_ECF	U_EECF	I_EECF	H_EECF
Case A	0.715063	0.736016	0.791213	0.765746	0.805471	0.806748	0.811454
Case B	0.786857	0.790401	0.798054	0.778801	0.791454	0.800019	0.805471
Case C	0.686848	0.737086	0.799012	0.764521	0.792534	0.803412	0.813306
Case D	0.735633	0.771238	0.801215	0.775512	0.802186	0.805471	0.811454
Case E	0.728105	0.763261	0.799011	0.774234	0.805471	0.813306	0.818675
Case F	0.742501	0.774087	0.793321	0.775021	0.793306	0.803412	0.810345
Case G	0.726196	0.764188	0.792042	0.779012	0.811454	0.809546	0.815471
Case H	0.712658	0.754164	0.786206	0.767911	0.805291	0.813306	0.821274

4.5Impact of parameters

Several parameters are used by the proposed technique and the traditional approaches. For achieving better accuracy, the value of these parameters is learned experimentally. This segment illustrates the impact of the parameters used to carry out the experiments.

4.5.1Similarity threshold

To compute the similarity among users/items, a similarity threshold is set to extract only those similar users/items that meet the required norms [39]. The parameters used for similarity computations are; γ, δ, θ *and* η. For significance weighting, the value of γ and δ is set to 30 and 25, respectively [53]. Furthermore, for finding the final similarity, the value 0.4 is assigned to both the parameters, θ and η [53]. The similarity threshold is used for the training of samples to tackle the sparsity problem.

4.5.2Significance of $\tilde{N}$

The impact of the number of hidden layers on the performance of the proposed approach is analyzed. For this purpose, the value of $\tilde{N}$ is varied from 5 to 35 with a step size of 5. The variations of N for different cases are depicted in *Figure 6* and *Figure 7* for ELM-based and EELM-based CF techniques, respectively. Consequently, the value of $\widetilde{N}$, where the proposed approach achieves the highest value of the F-measure, is considered.

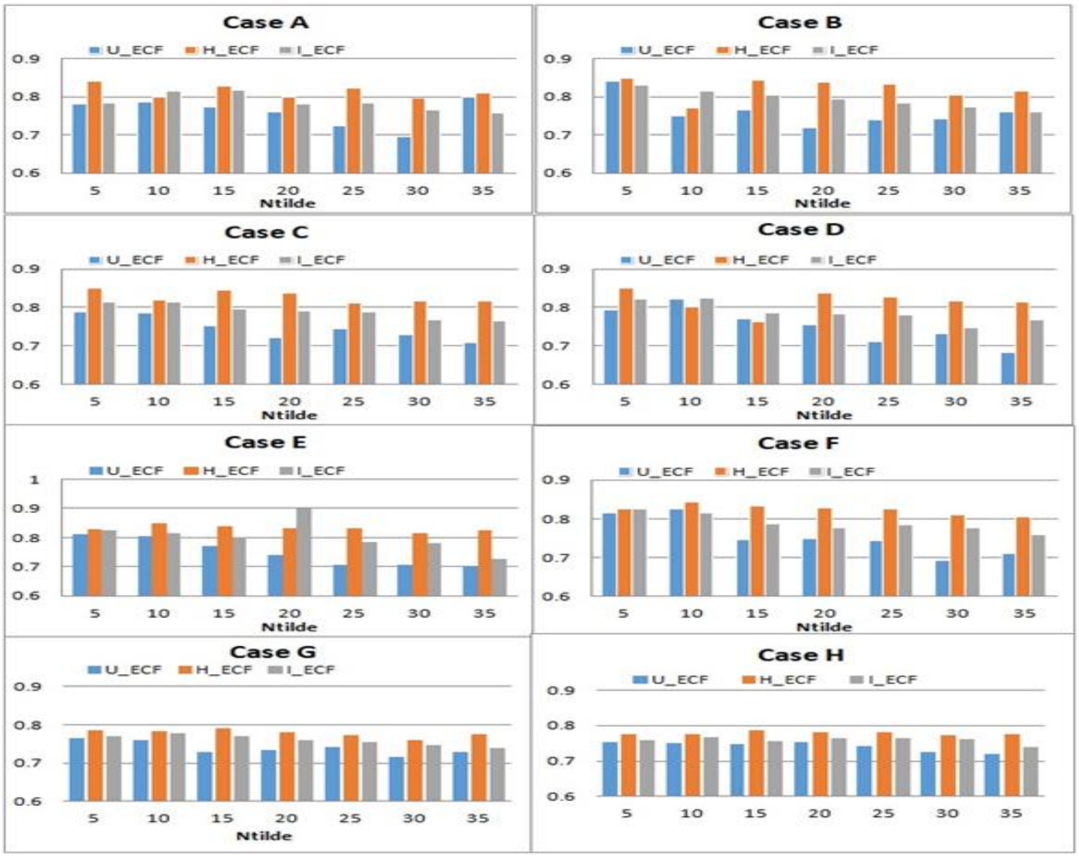

Figure 6 Impact of Ntilde on F-measure of ELM-based CF techniques

For the different cases of the RCF technique, the ELM/EELM-based variants of the proposed technique are observed for the highest value of the F-measure as shown in graphs depicted in *Figures 6* and *7.* Accordingly, for the highest value of F-measure, the corresponding value of $\widetilde{N}$ is considered for further experiments as given in *Table 7.*

4.5.3Impact of λ

The parameter λ decides the consequences of user based and item-based approaches on the hybrid recommendation model [24]. It ranges from 0 to 1, where 0 indicates that final predictions wholly rely on user-based model whereas value 1 specifies that recommendations are based entirely on an Item-based approach. For the experimental calculations, the value of λ is set to 0.7 [58]. In this way, all parameters are set empirically to achieve a better performance of the proposed technique.

5.Discussions

CF-based RSs have a major problem of sparsity which degrades the quality of recommendations. The proposed approach handles the problem of sparsity of CF and generates more accurate recommendations. This model captures the efficacy of the EMDP model [58] to fill the missing values in the rating dataset. Further, to reduce unnecessary computation, the proposed work drops the items receiving ratings by less than 20 users. Thereafter, applies user-based and item-based ELM models, U_ECF and I_ECF, motivated with the RCF model [39] to generate more accurate recommendations for an active user. From experimental results given in *Tables 3, 4, 5*, and *6*, it can be observed that the proposed ELM-based models, U_ECF and I_ECF improve the accuracy of prediction when compared with EMDP and RCF techniques. ELM-based schemes randomly select the parameters: weight and bias and the performance of these schemes depends upon the choice of these parameters.

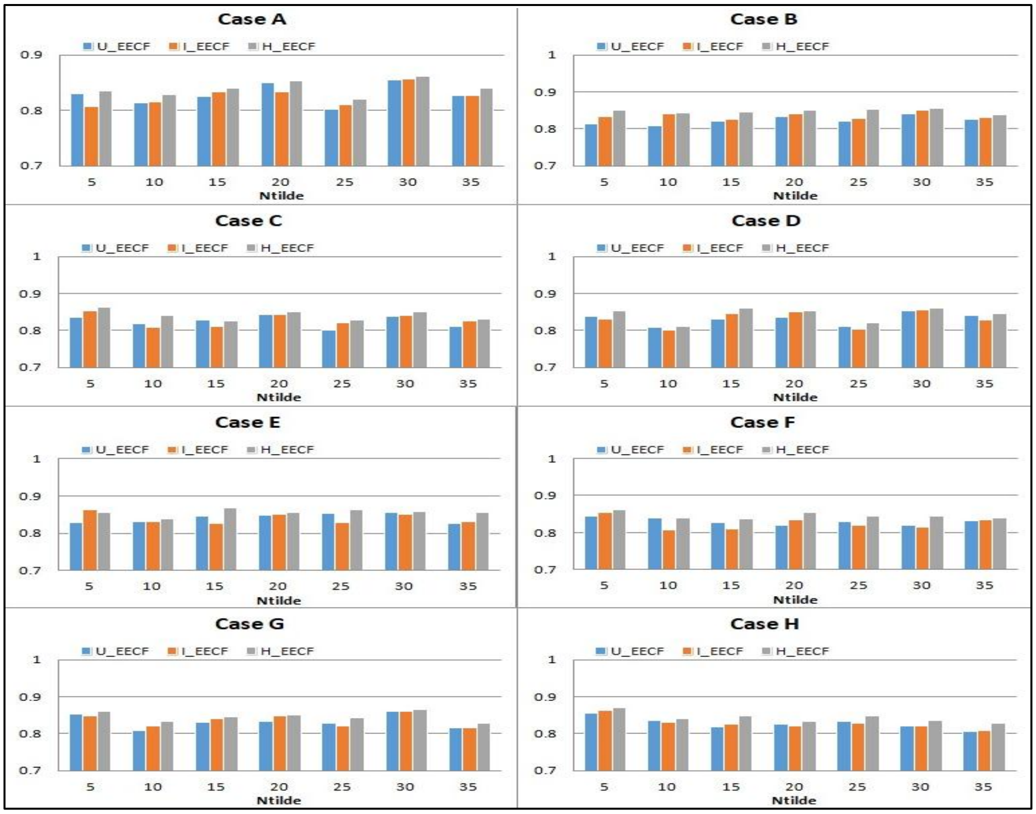

Figure 7 Impact of Ntilde on F-measure of EELM-based CF Techniques

Table 7 Value of N-tilde for proposed technique

Technique	Case A	Case B	Case C	Case D	Case E	Case F	Case G	Case H
U_ECF	10	5	5	10	5	10	5	5
I_ECF	15	5	5	10	5	5	10	10
H_ECF	5	5	5	5	10	10	15	15
U_EECF	30	30	20	30	30	5	30	5
I_EECF	30	30	5	30	5	5	30	5
H_EECF	30	30	5	30	15	5	30	5

Consequently, an evolutionary algorithm is applied with ELM-based schemes that effectively learns the value of parameters and improves performance. With the application of the evolutionary algorithm with ELM (named EELM), we developed U_EECF and I_EECF schemes. From the experimental results shown in *Tables 3, 4, 5*, and *6*, it is noticed that the EELM-based variants of the proposed model further enhance the quality of recommendations of the proposed variants of the ELM model. Moreover, user-based and item-based schemes generate different recommendations with different accuracy which can complement each other. To capture this phenomenon, we tried to combine the recommendations generated by both schemes. Therefore, we developed hybrid schemes H_ECF and H_EECF by combining the predictions made by respective User-based and Item-based approaches. From the results given in *Tables 3*, *4*, *5*, and *6*, we observed that the hybrid schemes H_ECF and H_EECF increase the prediction accuracy of respective User-based and Item-based models.

For observing the performance of the proposed variants, we considered both prediction and classification accuracy. Variants of proposed approach are compared with EMDP based on prediction accuracy on four different samples of the dataset. The accuracy metrics MAE and RMSE, on average, were decreased by factors of 9.69% and 32.83%, respectively, using the proposed technique. Thereafter, different cases discussed in [58] are considered and the classification accuracy of proposed variants is compared with that of the RCF model. From the results given in *Tables 4, 5*, and *6*, it is computed that the proposed scheme improves the classification accuracy of RS by a factor of 11.54% on average.

The variants of the proposed approach employ various parameters. For similarity computations, the parameters used are; γ,δ,θ and η which are set using the state-of-the-art approaches [53, 39]. The ELM and EELM-based approaches require several hidden layers, $\tilde{N}$, to train the model. How the value of $\tilde{N}$ impacts the performance of the proposed approach is studied by varying its value from 5 to 35 as shown in *Figures 6* and *7*. Based on this observation, the value of $\tilde{N}$ is selected for the best performance of the proposed model and used for further experiments as given in *Table 7*. Another parameter λ is used for balancing the predictions made by user-based and item-based approaches. The value of this parameter is set to be 0.7 based on the approach suggested in [58].

Limitations

The proposed approach has a significant limitation in that it requires the learning of multiple parameters. This training process adds computational overhead and effort. However, not tuning the parameter values can impact the results. Furthermore, the online processing steps involved are computationally expensive. In addition to addressing the problems of sparsity and accuracy, our approach fails to consider other important issues such as scalability, diversity, and cold-start. These aspects should be considered for a comprehensive recommender system solution.

A list of acronyms used in the paper is given in *Appendix I.*

6.Conclusion and future work

In this work, an EELM-based recommendation engine was designed and developed within the framework of CF to address the problem of sparsity and improve the accuracy of user preference prediction. The performance of the proposed method variants was compared with traditional recommendation systems, and various metrics such as MAE, RMSE, precision, recall, and F-measure were employed for analysis. Based on the experimental analysis, it was observed that the variants of the proposed scheme achieved higher accuracy in predictions. It should be noted that the focus of the proposed work was solely on a single feature, namely the rating of items.

In future research, the incorporation of other features, such as information about items like genres, director, release date, and users' personal information, is desired. The inclusion of multiple features in the database would enhance its usefulness, but it would also necessitate the development of feature selection procedures to address scalability challenges. Moreover, the success of social RSs in exploring social network information has led to the issue of information overload and interaction overload. To tackle these challenges, exploring how the ELM model can be employed within the framework of social RSs represents a potential direction for future work. Furthermore, the effective use of EELM in the domain of Content-based filtering is another area that warrants further research to address associated issues and enhance prediction accuracy. Additionally, it has been observed that RSs achieve improved accuracy by incorporating sentiment analysis. Hence, a new research direction would involve incorporating and analyzing sentiment analysis with ELM in the domain of RSs. By exploring these future directions, the capabilities and performance of recommendation systems can be enhanced, resulting in more accurate and effective user recommendations.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Author’s contribution statement

Pratibha Yadav: Worked on conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing original draft, reviewing, editing, visualization. **Shweta Tyagi:** Worked on conceptualization, writing original draft, reviewing, editing, visualization. **Harmeet Kaur:** Worked as a supervisor on conceptualization, writing, reviewing, editing, and project administration.

References

[1] Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J. Grouplens: an open architecture for collaborative filtering of netnews. In proceedings of the 1994 ACM conference on computer supported cooperative work 1994 (pp. 175-86).

[2] Shardanand U, Maes P. Social information filtering: algorithms for automating “word of mouth”. In proceedings of the SIGCHI conference on human factors in computing systems 1995 (pp. 210-7).

[3] Adomavicius G, Tuzhilin A. 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; 17(6):734-49.

[4] Bobadilla J, Ortega F, Hernando A, Gutiérrez A. Recommender systems survey. Knowledge-Based Systems. 2013; 46:109-32.
[5] De CR, Van HL, Htun NN, Štiglic G, Vanden AV, Verbert K. Health recommender systems: systematic review. Journal of Medical Internet Research. 2021; 23(6):1-21.
[6] Behera G, Nain N. Collaborative filtering with temporal features for movie recommendation system. Procedia Computer Science. 2023; 218:1366-73.
[7] Xu C, Ding AS, Liao SS. A privacy-preserving recommendation method based on multi-objective optimisation for mobile users. International Journal of Bio-Inspired Computation. 2020; 16(1):23-32.
[8] Sood M, Kaur H. Preference based personalized news recommender system. International Journal of Advanced Computer Research. 2014; 4(2):575-81.
[9] Li G, Zhang J. Music personalized recommendation system based on improved KNN algorithm. In IEEE 3rd advanced information technology, electronic and automation control conference 2018 (pp. 777-81). IEEE.
[10] Wang Y, Ma W, Zhang M, Liu Y, Ma S. A survey on the fairness of recommender systems. ACM Transactions on Information Systems. 2023; 41(3):1-43.
[11] Bedi P, Kaur H. Trust based personalized recommender system. INFOCOMP Journal of Computer Science. 2006; 5(1):19-26.
[12] Koren Y, Rendle S, Bell R. Advances in collaborative filtering. Recommender Systems Handbook. 2021: 91-142.
[13] Balabanović M, Shoham Y. Fab: content-based, collaborative recommendation. Communications of the ACM. 1997; 40(3):66-72.
[14] Tyagi S, Bharadwaj KK. A hybrid recommender system using rule-based and case-based reasoning. International Journal of Information and Electronics Engineering. 2012; 2(4):586-90.
[15] Walek B, Fajmon P. A hybrid recommender system for an online store using a fuzzy expert system. Expert Systems with Applications. 2023; 212:118565.
[16] Lops P, De GM, Semeraro G. Content-based recommender systems: state of the art and trends. Recommender Systems Handbook. 2011:73-105.
[17] Çano E, Morisio M. Hybrid recommender systems: a systematic literature review. Intelligent Data Analysis. 2017; 21(6):1487-524.
[18] Schafer JB, Frankowski D, Herlocker J, Sen S. Collaborative filtering recommender systems. The Adaptive Web: Methods and Strategies of Web Personalization. 2007: 291-324.
[19] Liang S, Ma L, Yuan F. A singularity-based user similarity measure for recommender systems. International Journal of Innovative Computing Information and Control. 2015; 11(5):1629-38.
[20] Fkih F. Similarity measures for collaborative filtering-based recommender systems: review and experimental comparison. Journal of King Saud University-Computer and Information Sciences. 2022; 34(9):7645-69.
[21] Jeong B, Lee J, Cho H. Improving memory-based collaborative filtering via similarity updating and prediction modulation. Information Sciences. 2010; 180(5):602-12.
[22] Pennock DM, Horvitz EJ, Lawrence S, Giles CL. Collaborative filtering by personality diagnosis: a hybrid memory-and model-based approach. Proceedings of the sixteenth conference on uncertainty in artificial intelligence 2010 (pp. 473-80). ACM.
[23] Chang TM, Hsiao WF, Chang WL. An ordinal regression model with SVD hebbian learning for collaborative recommendation. Journal of Information Science and Engineering. 2014; 30(2):387-401.
[24] Yadav P, Tyagi S. Hybrid fuzzy collaborative filtering: an integration of item-based and user-based clustering techniques. International Journal of Computational Science and Engineering. 2017; 15(3-4):295-310.
[25] Rodpysh KV, Mirabedini SJ, Banirostam T. Employing singular value decomposition and similarity criteria for alleviating cold start and sparse data in context-aware recommender systems. Electronic Commerce Research. 2023; 23(2):681-707.
[26] Kunegis J, Albayrak S. Adapting ratings in memory-based collaborative filtering using linear regression. In international conference on information reuse and integration 2007 (pp. 49-54). IEEE.
[27] Shang T, He Q, Zhuang F, Shi Z. Extreme learning machine combining matrix factorization for collaborative filtering. In the international joint conference on neural networks 2013 (pp. 1-8). IEEE.
[28] Zhu H, Luo Y, Weng C, Li M. A collaborative filtering recommendation model using polynomial regression approach. In fourth China grid annual conference 2009 (pp. 134-8). IEEE.
[29] Brenner A, Pradel B, Usunier N, Gallinari P. Predicting most rated items in weekly recommendation with temporal regression. In proceedings of the workshop on context-aware movie recommendation 2010 (pp. 24-7). ACM.
[30] Frank E, Hall M. Additive regression applied to a large-scale collaborative filtering problem. In AI: advances in artificial intelligence: 21st Australasian joint conference on artificial intelligence Auckland, New Zealand, 2008 (pp. 435-46). Springer Berlin Heidelberg.
[31] Montanés E, Quevedo JR, Díaz I, Ranilla J. Collaborative tag recommendation system based on logistic regression. ECML PKDD Discovery Challenge. 2009: 173-88.
[32] Park ST, Chu W. Pairwise preference regression for cold-start recommendation. In proceedings of the third conference on recommender systems 2009 (pp. 21-8). ACM.
[33] Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications. Neurocomputing. 2006; 70(1-3):489-501.

[34] Lika B, Kolomvatsos K, Hadjiefthymiades S. Facing the cold start problem in recommender systems. Expert Systems with Applications. 2014; 41(4):2065-73.
[35] Mild A, Natter M. Collaborative filtering or regression models for internet recommendation systems?. Journal of Targeting, Measurement and Analysis for Marketing. 2002; 10:304-13.
[36] Vucetic S, Obradovic Z. Collaborative filtering using a regression-based approach. Knowledge and Information Systems. 2005; 7:1-22.
[37] Chuan Y, Jieping X, Xiaoyong DU. Recommendation algorithm combining the user-based classified regression and the item-based filtering. In proceedings of the 8th international conference on electronic commerce: the new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet 2006 (pp. 574-8). ACM.
[38] Purushotham S, Liu Y, Kuo CC. Collaborative topic regression with social matrix factorization for recommendation systems. International conference on machine learning 2012 (pp. 1-8).
[39] Deng W, Zheng Q, Chen L. Real-time collaborative filtering using extreme learning machine. In international joint conference on web intelligence and intelligent agent technology 2009 (pp. 466-73). IEEE.
[40] Huang GB, Zhou H, Ding X, Zhang R. Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2011; 42(2):513-29.
[41] Wang Y, Cao F, Yuan Y. A study on effectiveness of extreme learning machine. Neurocomputing. 2011; 74(16):2483-90.
[42] Huang GB. Learning capability and storage capacity of two-hidden-layer feedforward networks. IEEE Transactions on Neural Networks. 2003; 14(2):274-81.
[43] Tyagi S, Yadav P, Arora M, Vashisth P. Predicting users' interest through ELM-based collaborative filtering. Data, Engineering and Applications. 2019:23-33.
[44] Blum K. Density matrix theory and applications. Springer Science & Business Media; 2012.
[45] Song S, Wang M, Lin Y. An improved algorithm for incremental extreme learning machine. Systems Science & Control Engineering. 2020; 8(1):308-17.
[46] Alencar AS, Neto AR, Gomes JP. A new pruning method for extreme learning machines via genetic algorithms. Applied Soft Computing. 2016; 44:101-7.
[47] Feng G, Huang GB, Lin Q, Gay R. Error minimized extreme learning machine with growth of hidden nodes and incremental learning. IEEE Transactions on Neural Networks. 2009; 20(8):1352-7.
[48] Zhang W, Jiang Q, Chen L, Li C. Two-stage ELM for phishing web pages detection using hybrid features. World Wide Web. 2017; 20:797-813.
[49] Scardapane S, Comminiello D, Scarpiniti M, Uncini A. Online sequential extreme learning machine with kernels. IEEE Transactions on Neural Networks and Learning Systems. 2014; 26(9):2214-20.
[50] Zhu QY, Qin AK, Suganthan PN, Huang GB. Evolutionary extreme learning machine. Pattern Recognition. 2005; 38(10):1759-63.
[51] Cao J, Lin Z, Huang GB, Liu N. Voting based extreme learning machine. Information Sciences. 2012; 185(1):66-77.
[52] Sánchez-monedero J, Gutiérrez PA, Hervás-martínez C. Evolutionary ordinal extreme learning machine. In hybrid artificial intelligent systems: 8th international conference, Salamanca, Spain 2013 (pp. 500-9). Springer Berlin Heidelberg.
[53] Savitha R, Suresh S, Sundararajan N. Fast learning circular complex-valued extreme learning machine (CC-ELM) for real-valued classification problems. Information Sciences. 2012; 187:277-90.
[54] Liu X, Li P, Gao C. Symmetric extreme learning machine. Neural Computing and Applications. 2013; 22:551-8.
[55] De JK. Learning with genetic algorithms: an overview. Machine Learning. 1988; 3:121-38.
[56] Hazir E, Ozcan T, Koç KH. Prediction of adhesion strength using extreme learning machine and support vector regression optimized with genetic algorithm. Arabian Journal for Science and Engineering. 2020; 45:6985-7004.
[57] Hahnloser RH, Sarpeshkar R, Mahowald MA, Douglas RJ, Seung HS. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature. 2000; 405(6789):947-51.
[58] Ma H, King I, Lyu MR. Effective missing data prediction for collaborative filtering. In proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval 2007 (pp. 39-46). ACM.
[59] Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. In proceedings of the 27th international conference on machine learning 2010 (pp. 807-14).
[60] Herlocker JL, Konstan JA, Terveen LG, Riedl JT. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS). 2004; 22(1):5-3.
[61] Karypis G. Evaluation of item-based top-n recommendation algorithms. In proceedings of the tenth international conference on information and knowledge management 2001 (pp. 247-54). ACM.
[62] Kim TH, Yang SB. An effective threshold-based neighbor selection in collaborative filtering. In advances in information retrieval: 29th European conference on IR research, Rome, Italy, 2007 (pp. 712-5). Springer Berlin Heidelberg.

Pratibha Yadav was born in Delhi, India in 1989. She received her MCA degree from University of Delhi, New Delhi, India in 2014. She joined Shyama Prasad Mukherji College, University of Delhi, New Delhi, India as an Assistant Professor in 2014. She is currently pursuing her Phd in Computer Science from University of Delhi. Her research interest includes Information Retrieval, Machine Learning, Artificial Intelligence and Knowledge Discovery in Databases (KDD).
Email: pratibha.123@spm.du.ac.in

Shweta Tyagi was born in Meerut, India in 1980. She received her MSc degree in Mathematics from the Indian Institute of Technology, Delhi, India in 2003. She got her M.Tech. in 2008 and PhD in Computer Science in 2013 from Jawaharlal Nehru University, Delhi, India. She is currently an Assistant Professor at Shyama Prasad Mukherji College, University of Delhi, Delhi, India. She joined in 2013. Prior to joining Shyama Prasad Mukherji College, she worked as a faculty member in the Computer Science Department at IITM, Delhi, India. Her research interest includes Machine Learning, Artificial Intelligence, Soft Computing and Information Retrieval.
Email: shwetakaushik2006@gmail.com

Harmeet Kaur was born in Delhi, India in 1973. She completed her Ph.D. in 2007 from the University of Delhi, India. She is an Associate Professor in the Department of Computer Science, Hansraj College, University of Delhi. Her teaching experience spans over a period of around 24 years. Her research interests lie in the field of Recommender Systems and Crowdsourcing. She has published around 45 research papers in national and international journals and conferences of repute.
Email: hkaur@hrc.du.ac.in

Appendix I

S. No.	Abbreviation	Description
1	CF	Collaborative Filtering
2	EELM	Evolutionary Extreme Learning Machine
3	ELM	Extreme Learning Machine
4	EMDP	Effective Missing Data Prediction
5	GA	Genetic Algorithm
6	GPS	Global Positioning System
7	H_ECF	Hybrid of U_ECF and I_ECF
8	H_EECF	Hybrid of U_EECF and I_EECF
9	I_ECF	Item-based ELM model for CF
10	I_EECF	Item-based Evolutionary ELM model for CF
11	IoT	Internet of Things
12	MAE	Mean Absolute Error
13	RCF	Real-time CF
14	ReLU	Rectified Linear Unit
15	RFID	Radio Frequency Identification
16	RMSE	Root Mean Square Error
17	RS	Recommender System
18	SLFN	Single-Hidden Layer Feed-Forward Neural Network
19	SVM	Support Vector Machine
20	U_ECF	User-based ELM model for CF
21	U_EECF	User-based Evolutionary ELM model for CF