

# A Comprehensive Collaborating Filtering Approach using Extended Matrix Factorization and Autoencoder in Recommender System

Mahamudul Hasan<sup>1</sup>

Department of Computer Science and Engineering  
East West University  
Dhaka, Bangladesh

Falguni Roy<sup>2</sup>

Institute of Information Technology,  
Noakhali Science and Technology University  
Noakhali, Bangladesh

Tasdikul Hasan<sup>3</sup>

Department of Computer Science and Engineering  
East West University  
Dhaka, Bangladesh

Lafifa Jamal<sup>4</sup>

Department of Robotics and Mechatronics Engineering  
University of Dhaka  
Dhaka, Bangladesh

**Abstract**—Recommender system is an approach where users get suggestions based on their previous preferences. Nowadays, people are overwhelmed by the huge amount of information that is being present in any system. Sometimes, it is difficult for a user to find an appropriate item by searching the desired content. Recommender system assists users by providing suggestions of required information or items based on the similar features among the users. Collaborative filtering is one of the most re-known process of recommender system where the recommendation is done by similar users or similar items. Matrix factorization is an approach which can be used to decompose a matrix into two or more matrix to generate features. Again, autoencoder is a deep learning based technique which is used to find hidden features of an object. In this paper, features are calculated using extended matrix factorization and autoencoder and then a new similarity metric has been introduced that can calculate the similarity efficiently between each pair of users. Then, an improvement of the prediction method is introduced to predict the rating accurately by using the proposed similarity measure. In the experimental section, it has been shown that our proposed method outperforms in terms of mean absolute error, precision, recall, f-measures, and average reciprocal hit rank.

**Keywords**—Recommender system; deep learning; autoencoder; matrix factorization; similarity measures

## I. INTRODUCTION

Recommender system is introduced as an assistant for the users to obliterate information overload problem of the internet. It finds the best information for the users by identifying the pattern from the dataset and then recommends it to the users. The real life examples of recommender system can be seen on most of the websites like YouTube, Amazon where it recommends videos or products. In those websites, two users are selected as similar when both follow the same feature pattern. Content-based and collaborative filtering are the two approaches in the field of recommender system [1]. In collaborative filtering, the behavior of the similar users are being analyzed and on the basis of the analysed result, the recommendation has been done to the target users. On the basis of similarity measures' methodology, collaborative

filtering further divided into two category, one is memory-based approach and another one is model-based approach [2]. Again, memory-based collaborative filtering could be sub categorised as user-based approach and item-based approach [3]. In user-based collaborative filtering approach, products are recommended to a user which have been liked by other similar users [4]. On the other hand in item-based collaborative filtering, items are recommended according to the similarity of the products or on the basis of the ratings of the user on a similar product which has been rated previously [5].

Many sites like Amazon, Pandora and Netflix are using recommender system for optimizing performance, making more profit and also for boosting sales [6]. It is not only bringing profit to the sites but also creating public trusts for those sites. Recommender system already has been captured most of the space of computer science. As, it is a multi-disciplinary field so analysis and exploration can be done for this field in the context of information mining, machine learning, human-computer interaction, social statistics, network analysis, distributed and mobile systems, artificial societies, computational trust, etc. Researchers are now exploring how the content information can be used to calculate the recommendations.

User behavior is a very complex function so it would be very difficult to find the similarity between users [7]. In this research, a user-based collaborative filtering approach has been employed with matrix factorization and autoencoder. Matrix factorization can provide better performance in predicting and finding the similarity between users. Again, autoencoder is also a very useful technique for satisfactory recommendation. Combination of this two techniques have been implemented in this paper with good performances. So in this paper, the features for the users have been extracted by using the Autoencoder (AE) [8] and the Extended Matrix Factorization (EMF) [9]. After that, the similarity between the users have been calculated and by using the similarity, a recommendation prediction algorithm have also been proposed to accelerate the performances of the recommender system. Finally, the performance of our system are shown in the result section

to evaluate the final recommender system accuracy.

## II. BACKGROUND STUDY AND RELATED WORKS

In the recommender system, there are many methods which are used to provide better performance and deep learning technique is one of them [10]. Sometimes, it can provide the best solution where others method may fail to do so [11]. It can be used as a solution of multiple problems like, speech recognition, image processing and recognition, natural language processing etc. Few types of deep learning methods are discussed here which are being used in most of the recommender system.

### A. Multilayer Perceptron

Multilayer perceptron (MLP) is a kind of feedforward neural network which has multiple hidden layers. There can be more than linear layers or neurons in the MLP [12]. It is like the basic structure of the neural network when three layers are taken where the input layer is the first layer, the hidden layer is the second layer and the output layer is the last layer. The structure of the MLP is represented in Fig. 1. MLPs can be construed as stacked nonlinear transformation layers and layers can be taken as much as needed for the sake of computation.

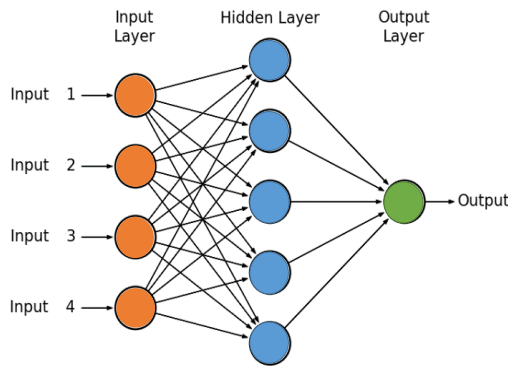


Fig. 1. Multilayer Perceptron.

### B. Convolutional Neural Networks

Mostly convolutional neural networks (CNN) are ideal for image processing. Here, Fig. 2 is an example of convolutional neural network [13]. CNN is normally designed for reduced processing requirements otherwise this it is same as the simple multilayer perceptron. In the hidden layers of the CNN, multiple convolution layers, normalization layers, and pooling layers are utilized [14].

### C. Recurrent Neural Networks

Recurrent neural network (RNN) is designed to recognize the sequential characteristics of the data. They are designed with the loop or they are neural networks with loop and that helps data to preserve it in its memory [15]. They have a specialty that they use the feedback loops to process the data sequences. It can process data into output but the difference is that it can use feedback loops throughout the process unlike other regular feedforward system. These things help RNN to process sequential data. A visual comparison of RNN and traditional feedforward network is shown in Fig. 3.

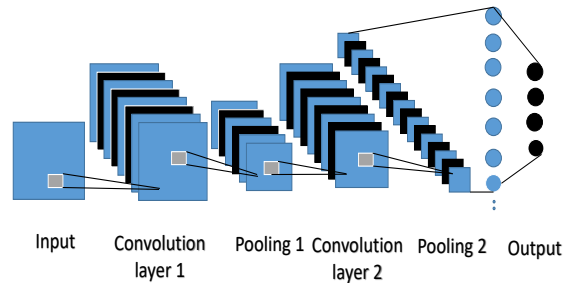


Fig. 2. Convolutional Neural Networks.

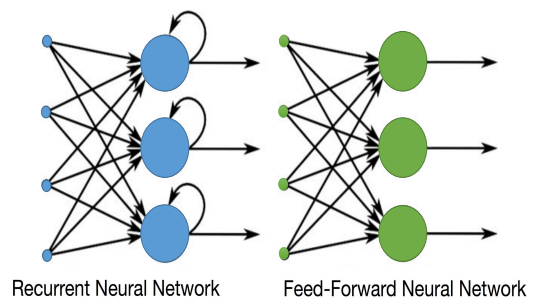


Fig. 3. Comparative Image of RNN and Traditional Feedforward Network.

### D. Restricted Boltzmann Machines

A Restricted Boltzmann Machine (RBM) may be a generative stochastic falsified neural arrangement capable of learning a likelihood of transmission over its set of inputs [16]. Boltzmann Machines (BMs) are a particular type of Markov Random Field (MRF) log-straight. Boltzmann Machines further confine BMs to those without noticeable unmistakable and covered up concealed associations. The energy function  $P(x, y)$  of a RBM is defined as follows:

$$P(x, y) = -a'x - b'y - b'Wu \quad (1)$$

here,  $W$  represents the weights connecting hidden and visible units and  $a, b$  are the offsets of the visible and hidden layers, respectively.

### E. Principal Component Analysis

Simplification of complexity in high dimensional data is done by principal component analysis (PCA). PCA perform its work by transforming data into fewer dimension [17]. PCA is an unsupervised learning and also similar to the clustering method. PCA lowers data by expressing it geometrically on significantly lower dimensions called main components (PCs), with the aim of choosing the best overview of data by using a limited amount of PCs. PCA gets a new set of configurations (or a set of views) so that all the measurements are orthogonal (and therefore linearly independent) and actually ranked

according to the variance between the data. PCA works by measuring the data point covariance matrix and by calculating its eigen vectors and their eigen values.

#### F. Singular Value Decomposition

The singular-value decomposition (SVD) is a factorization of a real or complex matrix [18] and it decomposes a matrix into three other matrices that are shown in equation 2.

$$A = U\Sigma V^T \quad (2)$$

where,  $A$  is an  $m \times n$  matrix,  $U$  is an  $m \times m$  orthogonal matrix,  $S$  is an  $n \times n$  diagonal matrix,  $V$  is an  $n \times n$  orthogonal matrix. By using the complex features of SVD, recommendations are made to the users.

#### G. Collaborative Filtering

Collaborative filtering (CF) is the most popular technique in the field of recommender system. It uses algorithms to find out the similar users or items with similar preferences and characteristics to make a personalized recommendation [19]. User-based CF calculates similarity between each pair of users which is used for recommendation. Again, item-based CF is another similarity based approach where similarities are calculated between items and then similar items are recommended to the users for the purpose of recommendation [20]. Suppose, there are  $m$  users and  $n$  items in a system in which the  $m \times n$  is a matrix that is signified the past behavior of the users. Each cell in the matrix represents the user-item association or interaction. For example, this signifies how user  $u$  prefers item  $m$ . There are two sorts of suppositions, express sentiment and a certain feeling. The preceding one explicitly indicates how a customer rates a item (consider rating an application or a film), although the last one just fills in as an intermediary giving us heuristics about how a customer loves a item (e.g. a number of preferences, clicks, visits) [21]. Express supposition is simpler than testable supposition because we do not have to think about what the number indicates. For example, there may be a tune that is particularly loved by the user, yet he once tunes in to it in view of the fact that he was occupied while tuning in to it. Without express sentiment, it can't be made sure whether the client loathes that thing or not. Notwithstanding, a large portion of the criticism that is gathered from users are understood.

Some collaborative filtering approaches are discussed here.

1) *Clustering Method using Classification Algorithm*: One of the issue of recommender system is the cold start issue which is identified with the existence of new users and items [22]. Recommender system does not have adequate data to make recommendations at the time of new users arrival [23]. This framework uses a technique of three arrangements to make suggestions for new users. Users are called as neighbors with the greatest amount of closeness to new users.

2) *Centroid-based Clustering Method Oriented*: The calculation created for centroid-based grouping is used for personalized recommender systems. This strategy includes two suggestion stages [24]. In the first stage, user's remarks are collected and in the second stage, suggestions for dynamic

users are finished by selecting the best quality groups. This framework solves many issues like cold start problem.

3) *Hierarchical Cluster Method*: For the suggestion, this strategy is used. Various leveled grouping is a technique of research to fabricate group progression [25]. It comprises of two types, agglomerative and disruptive. In this part, individual user profiles move towards similar user profiles and clients are isolated into a few gatherings in terms of closeness.

4) *Deep Semantic Similarity Model*: Deep semantic similarity model (DSSM) represents a broader model of semantic similarity [26]. DSSM, created by the MSR Deep Learning Technology Center, is a profound neural system (DNN) displaying strategy for speaking to content strings (sentences, inquiries, predicates, substance specifics, and so on.) in a consistent semantic space and demonstrating semantic similitude between two content strings [27]. DSSM has wide applications including data recovery and web look positioning, promotion choice/significance, logical substance hunt, and intriguing quality undertakings, question noting, information derivation, picture, and machine interpretation. DSSM can be utilized to create idle semantic models that venture elements of various kinds (e.g., inquiries and archives) into a typical low-dimensional semantic space for an assortment of machine learning assignments, for example, positioning and order. For instance, in web seek positioning, the significance of a report given a question can be promptly figured as the separation between them in that space.

### III. PROPOSED SYSTEM USING EXTENDED MATRIX FACTORIZATION AND AUTOENCODER

Usually, a recommender system's performance depends on the accuracy of similarity determination of users or items and this similarity is identified in an acceptable rate when there exists fairly dense amount of data in the data set. But at the case of new user or item, it is hard to predict the recommendation for that user or item because of the existence of the small amount of correlated data. The problem is considered as the cold start problem. Again, when the data set is sparse or when a new user or item enters in the system, there exists no similarity metrics that can be used. To eliminate these issues, a similarity measure have been proposed here. A flowchart has been displayed in Fig. 4 to depict our proposed recommender system.

#### A. User Similarity Determination by using Extended Matrix Factorization

Matrix factorization (MF) is a newly developed technique for finding parts-based data and linear representations. It has many sub categories with individual specialties but for the proposed system, non-negative matrix factorization (NNMF) has been used with some extension. NNMF is a group of logistic regression and linear algebra methodologies in which a  $T$  matrix is factorized into two  $P$  and  $Q$  matrices with the assumption that there exists no negative elements in all three matrices and an example can be seen from Fig. 5.

Formal consideration is given to extended matrix factorization (EMF) algorithm to solve the problem: extended matrix factorization given the non-negative matrix  $T$ , non-negative matrix factors  $P$  and  $Q$  as the following equations 3:

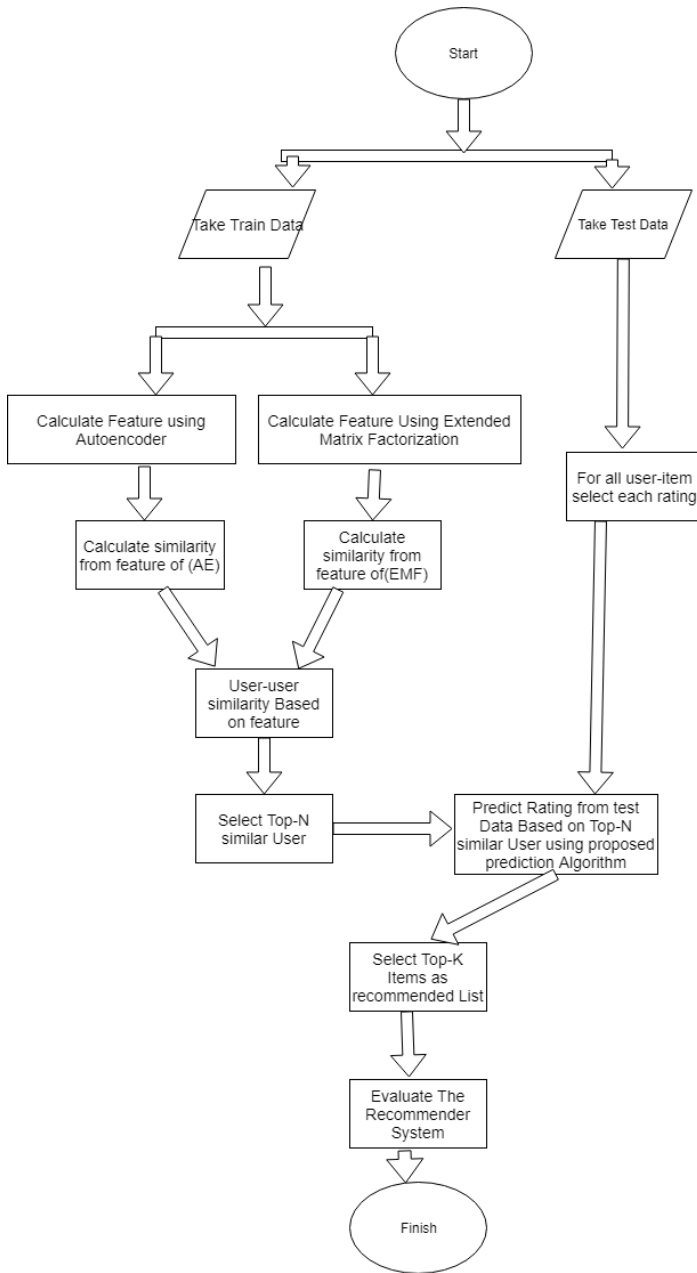


Fig. 4. Flowchart of the Proposed System.

$$T \sim P \times Q \quad (3)$$

In fact, these algorithms are very easy to implement and conversion properties are guaranteed which means continual update regulation iterations are pretty much guaranteed to converge an optimal local matrix factorization. For calculating similarity of the users, a user-item rating matrix has been taken where each column of the matrix denotes a movie and every row determine the system's users. For the calculation of the prediction of the rating of user-movie pair is done by using following equation 4:

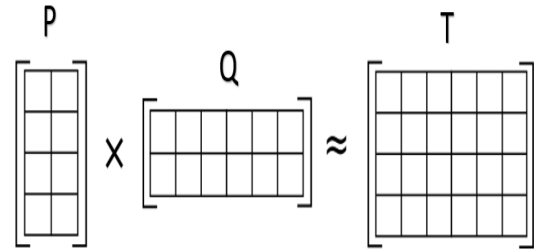


Fig. 5. Matrix Factorization.

$$\hat{X}_{ij} = \sum_{k=1}^k y_{ik} \times z_{kj} \quad (4)$$

After few iterations, it will generate predictions with errors. So get the desired output, the errors must be minimized with some function. So, gradient descent is a good option to minimize the errors and make the prediction more accurate as it helps converge to the minimum error. To calculate the errors, the following equation 5 have been used:

$$f_{ij} = (x_{ij} - \hat{x}_{ij}) = (x_{ij} - \sum_{k=1}^k y_{ik} \times z_{kj}) \quad (5)$$

To reduce the errors as small as possible and actually find the curve of the errors by taking the error gap for each element or value, the successive equations (equation 6 and 7) are used.

$$\frac{\delta}{\delta y_{ik}} = -2(x_{ij} - \hat{x}_{ij})(z_{kj}) = -2f_{ij}z_{kj} \quad (6)$$

$$\frac{\delta}{\delta z_{kj}} = -2(x_{ij} - \hat{x}_{ij})(y_{ik}) = -2f_{ij}y_{ik} \quad (7)$$

By using the learning rate, errors have to be updated. The learning rate determines that how far the gradient should be traveled. If the learning rate is not set properly, it might be set to a rate which does not overshoot the minimum and does not bring any unnecessary complexity. Equation 8 and 9 have been used for the updating of the errors.

$$y'_{ik} = y_{ik} + \alpha \frac{\delta}{\delta y_{ik}} f_{ij} = y_{ik} + 2\alpha f_{ij}z_{kj} \quad (8)$$

$$z'_{kj} = z_{kj} + \alpha \frac{\delta}{\delta z_{kj}} f_{ij} = z_{kj} + 2\alpha f_{ij}y_{ik} \quad (9)$$

At these stage, by subtracting the similarity of users, the difference between each pair of users can be found and after multiplying with  $-1$ , this difference turns into similarity.

$$Similar_{EMF}(u, v) = \sum_{k=1}^K (y_{(u,k)} - y_{(v,k)}) \times (-1) \quad (10)$$

After these calculation, the similarity of user-user can be found by using extended matrix factorization.

$$outL_i^j = \frac{1}{1 + e^{-L_i^j}} \quad (12)$$

### B. User Similarity Determination by using Autoencoder

Autoencoder (AE) is neural network that has the same outputs as the inputs. Its aim is to compress the input and rebuild the output from latent representation. This sort of neural network comprises of two parts, one being encoder and the other being decoder. An encoder is a major part of the neural network which mainly compresses the input into a latent representation. This can be presented with the  $l = f(x)$  encoding function. A decoder is a part meant for the restoration of the input representation of latent space. It can be defined with the function  $r = g(h)$  to decode. As the structure of the autoencoder is like a basic neural network so it gives the freedom to choose as much layers as needed on the basis of the requirement. For the proposed system, the rating has been considered as the input. So, the input layer has ratings as input data and the amount of input data is the same as the total dataset. The second layer contains 100 neurons and act as an encoder. The third layer is the latent feature representation layer which contains 50 neurons and the fourth layer contains 100 neurons also. Finally, the last layer contains the same number of neurons as the input layer. So, this autoencoder contains total five layers. All the layers and neurons are connected to each other. Their weights are also initialized. The main target of this stage is to find similarity between users by using autoencoder like in Fig. 6.

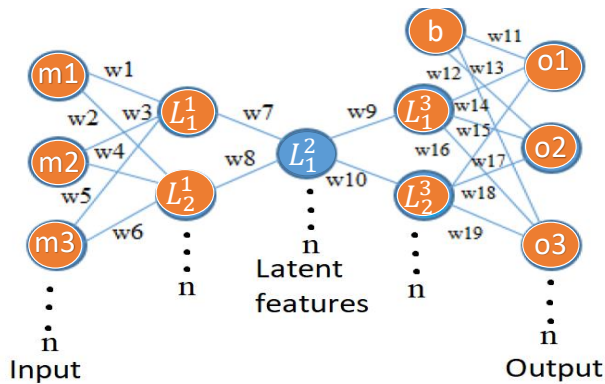


Fig. 6. Autoencoder with Five Layers.

Here,  $m_1, m_2, m_3$  to  $n$  denote the input neurons and synapses are initialized. For the computation value of the every neuron must be calculated. For calculating the neuron's value of the hidden layer, the ensuing equation 11 is used.

$$L_i^j = \sum m_i \times w_i \quad (11)$$

Here,  $L_i^j$  is the hidden layer where  $i$  is identified as the number of neurons and  $j$  is the number of hidden layers. For activation function, sigmoidal function has been used as the outputs of sigmoidal function exits between 0 and 1. After getting the value of  $L$ , it should be passed through the activation function to activate it. So here equation 12 is the activation function equation.

As all the neurons have the same structure and that's why calculation of the neurons till the output layer is the same with some added bias. After the calculation to the output layer, prediction can be found with some errors and to identify the errors following equation 13 has been used:

$$TotalF = \sum (m_i - o_i)^2 \quad (13)$$

At the stage to minimize and updating the errors, backpropagation is used which helps to minimize the errors and update the value of every neuron to make the correct prediction. The calculation of backpropagation for the output layers is as follows:

$$\frac{\delta TotalF}{\delta w_i} = \frac{\delta TotalF}{\delta outo_i} \times \frac{\delta outo_i}{\delta o_i} \times \frac{\delta o_i}{\delta w_i} \quad (14)$$

After backpropagation, errors will be reduced and these errors will help to update the weights and also produce accurate prediction. So, the updating equation:

$$w_i^+ = w_i + \alpha \delta w_i \quad (15)$$

The calculation of the backpropagation is not same for all hidden layers and different from the output layers also because the output of every hidden layer neurons contributes to several output neurons.

$$\frac{\delta TotalF}{\delta w_i} = \frac{\delta TotalF}{\delta outL_i^j} \times \frac{\delta outL_i^j}{\delta L_i^j} \times \frac{\delta L_i^j}{\delta w_i} \quad (16)$$

$$\frac{\delta TotalF}{\delta w_i} = (\delta_i \times w_i) \times outL_i^j \times i \quad (17)$$

$$\delta_i = (x_i - o_i) \times (outo_i) \times (1 - outo_i) \quad (18)$$

For these layers updating equation is the same which has been shown before. At this stage, the latent feature representation can be found at the third layer and after sufficient iterations, the prediction of the rating can be done. Afterward, subtracting the features of each user pair dissimilarity can be identified and multiplying it with  $-1$  user-user similarity is found. After that the similarities between users achieved from autoencoder and extended matrix factorization have been used for the prediction method.

$$Similar_{AE}(u, v) = \sum_{k=1}^K (L_{(u,k)} - L_{(v,k)}) \times (-1) \quad (19)$$

#### IV. PROPOSED PREDICTION METHOD

The prediction method which is described here is similar to the generic recommender system prediction method [28]. User based similarity that is achieved from both EMF and autoencoder, has been taken to make prediction more accurate. It can be called a comprehensive prediction method as two methods have been merged.

$$pred_{u,m} = \bar{r}_u + \frac{\sum_{v \in U} [Sim_{AE}(u,v) \times Sim_{EMF}(u,v) \times (r_{v,m} - \bar{r}_v)]}{\sum_{v \in U} Sim_{AE}(u,v) \times Sim_{EMF}(u,v)} \quad (20)$$

Here,  $\bar{r}_u$  is the average rating of the user  $u$  and  $r_{(v,m)}$  is the actual rating. Here,  $\bar{r}_v$  represents the average rating of the user  $v$ .

---

#### Algorithm 1 Proposed Prediction Function Algorithm.

---

INPUT: User( $u$ ), Item( $m$ ), Rating, Similarity  
OUTPUT: Predicted Rating

- 1:  $SU \leftarrow 0, TU \leftarrow 0, minRating \leftarrow 1, maxRating \leftarrow 5$
- 2: **for all**  $v \in Users$  **do**
- 3:     **if**  $ratings_{v,m} <> 0$  **then**
- 4:          $SU \leftarrow SU + sim(ae)_{u,v} \times sim(emf)_{u,v} \times (ratings_{v,m} - avgRat_v)$
- 5:          $TU \leftarrow TU + sim(ae)_{u,v} \times sim(emf)_{u,v}$
- 6:     **end if**
- 7: **end for**
- 8:  $userOffset \leftarrow (SU/TU)$
- 9:  $rating_{u,m} \leftarrow userOffset + avgRatings_u$
- 10: **if**  $(rating_{u,m} < minRating)$  **then**
- 11:      $rating_{u,m} \leftarrow minRating$
- 12: **end if**
- 13: **if**  $(rating_{u,m} > maxRating)$  **then**
- 14:      $rating_{u,m} \leftarrow maxRating$
- 15: **end if**
- 16: **if**  $(rating_{u,m} <> 0)$  **then**
- 17:     **Return**  $rating_{u,m}$
- 18: **end if**
- 19: **Return**  $(rating_{u,m})$

---

#### V. PERFORMANCE EVALUATION

Most of the existing recommender system's calculation is focused on the offline predictive accuracy assessment. The benefit of offline evaluation is that it is fast, cost-effective and can be used different data sets and metrics frequently. For our experimental purpose, offline analysis has been focused. The data set has experimented here is the Movielens (ML-1M) dataset. It contains a total user of 6040 and total movies of 3952 with a rating of 1000209 which is shown in Table I. Each user in the data set has been rated at least 20 movies. The density of the user-item matrix 4.1% in Movielens ML-1M data set. The important parameters of the data set are listed in the following Table I which are used in the experimental section.

The most commonly used evaluation methodologies are mean absolute error, precision, recall, f-measures, and average reciprocal hit rank. With regard to the system task, each metric has advantages and disadvantages. A true positive (TP) is an

TABLE I. IMPORTANT PARAMETERS OF THE DATASET.

Dataset	Users	Items	Rating
Movielens	6040	3952	1-5

upshot where the system accurately forecasts the positive class. Similarly, a true negative (TN) is an upshot where the system accurately forecasts the negative class. A false positive (FP) is an upshot where the system inaccurately forecasts the positive class. And a false negative (FN) is an upshot where the system inaccurately forecasts the negative class [29].

#### A. Mean Absolute Error

In the group of statistical accuracy metrics, the mean absolute error (MAE) compares the predicted ratings with the actual ratings [29]. MAE takes into consideration in particular the absolute average difference between the predicted rating and the user's actual rating which is shown in equation 21.

$$MAE = \frac{\sum_{i=1}^{m_a} |R_{a,i} - P_{a,i}|}{m_a} \quad (21)$$

However, evidence suggests that when the MAE is reduced, other metrics have shown significant improvement. In the field of recommender system, the good rating should be predicted as good so that good predicted item with good rating can be recommended. Mean absolute error also takes the bad rating into consideration. This shows that for the Top-N recommendations, MAE is not the best evaluation measure.

#### B. Precision

Precision actually defines that how precise a classification model is. When dealing with the classification, precision can be proved as a very useful evaluation measurement. It is the amount of correct positive prediction [30]. The proportion of recommended items which is precision that users in the test set really liked [31]. For better performance, this value should be high.

$$Precision = \frac{TP}{TP + FP} \quad (22)$$

#### C. Recall

The recall is the fraction of the relevant documents that are successfully retrieved [30]. value is well in the training set, which ranks the average percentage of components in the test set. For better performance, this value should be high.

$$Recall = \frac{TP}{TP + FN} \quad (23)$$

#### D. F-measure

Precision and recall are two of the important evaluation measurement and the weighted average of them is f-measure [32]. Good precision and bad recall results in bad recommendation. Again, bad precision and good recall results in bad recommendation as well. So, both effects should be higher for

better recommendation. F-measure value might be higher when both of the value of precision and recall is high is given in equation 24. Both precision and recall are taken into account to calculate the measure where precision is the ratio of right positive results divided by the number of positive outcomes and recall is the number of precise positive results divided by the number of positive outcomes.

$$F_{Measures} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (24)$$

### E. Average Reciprocal Hit Rank

The average reciprocal hit rank (ARHR) [33], [34] is a statistical measure of the ranking which offers a list of recommendations sorted by the probability of accuracy. The average reciprocal hit rank can be defined using the equation 25.

$$ARHR = \frac{1}{N} \times \sum_{i=1}^N \frac{1}{rank_i} \quad (25)$$

## VI. EXPERIMENT AND MEASUREMENT

Fig. 7 to 15 shows the results obtained with the data set of movielens 1 – m. From these figures, it has seen that how the proposed method and the quality measures (MAE, precision, recall, f - measures, ARHR) are carried out. Here, the results are generated using top-n neighbors vs precision, top-n neighbors vs recall, top-n neighbors vs f-measures, and top-k recommendation vs average reciprocal hit rank. With the increase of the neighbour size and recommendation significant improvement have been observed.

### A. Top-N Neighbor Vs Evaluation Measures

Mean absolute error, precision, recall, f-measures are utilized to evaluate the collaborative filtering recommender system. From Fig. 7 and 8, it can be seen that the relationship between neighbor size and MAE is inverse that is with the increase of the the neighbor size the mean absolute error will decrease. From Fig. 8, it is observed that the comprehensive approach of both autoencoder and extended matrix factorization (ACCFAERS) outperforms in all circumstances.

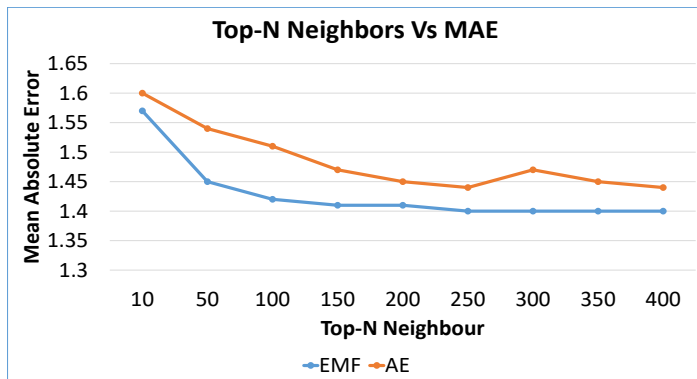


Fig. 7. Top-N Neighbor vs. mean absolute error for AE and EMF.

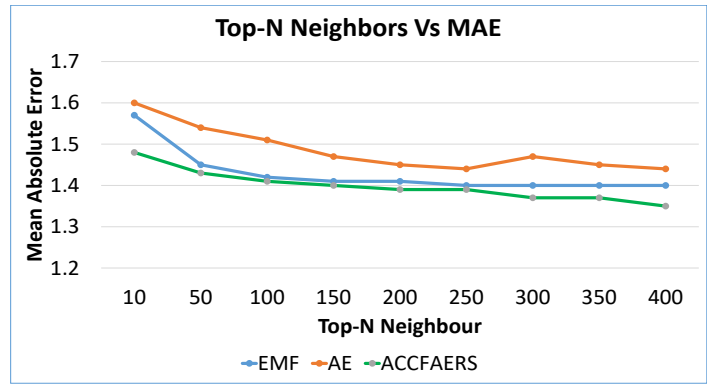


Fig. 8. Top-N Neighbor vs. mean absolute error for Proposed Method.

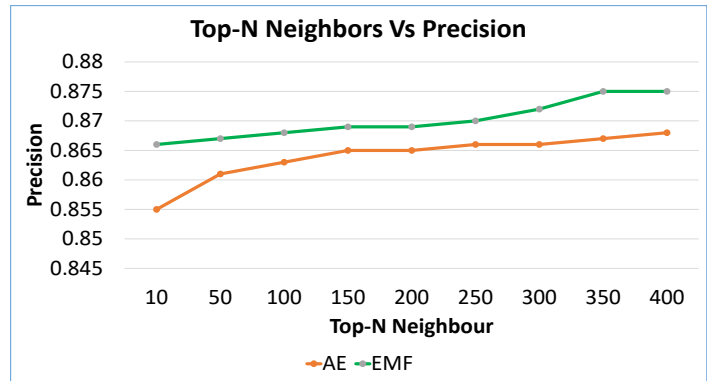


Fig. 9. Top-N Neighbor vs. Precision For AE and EMF.

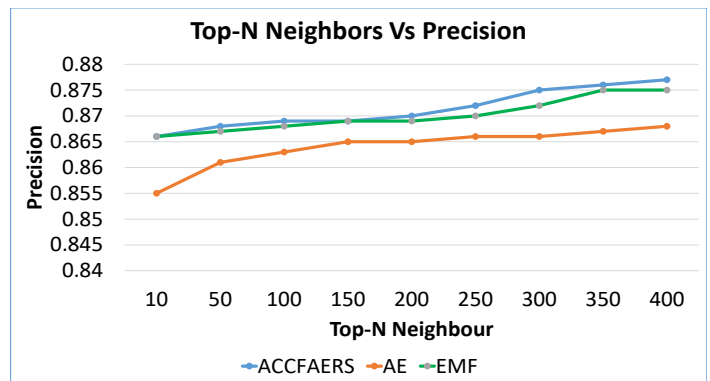


Fig. 10. Top-N Neighbor vs. Precision For Proposed method.

Again from Fig. 9 and 10, it is clear that the precision increases with the increase of neighbor size and the proposed method performs better than other methods.



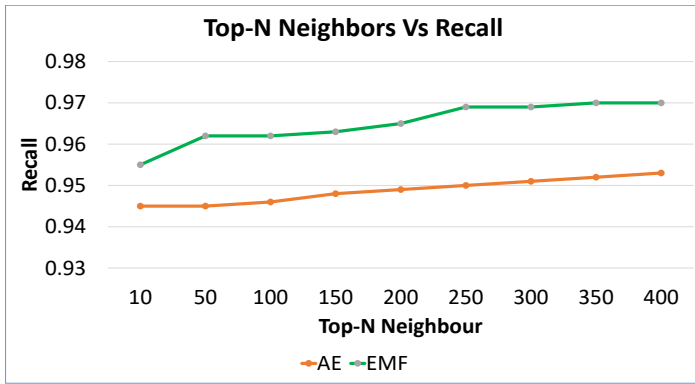


Fig. 11. Top-N Neighbor vs. Recall For AE and EMF.

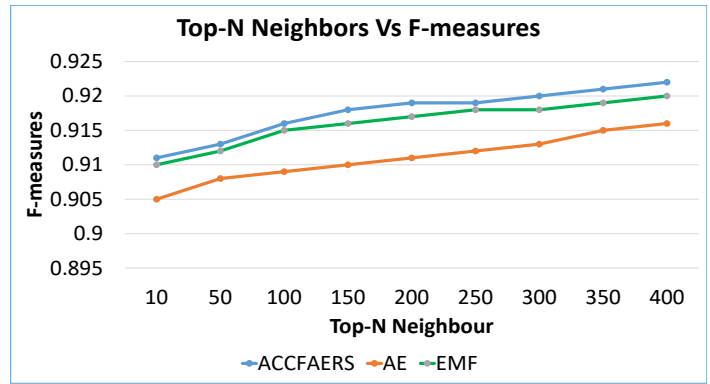


Fig. 14. Top-N Neighbor vs. F-measures For Proposed method.

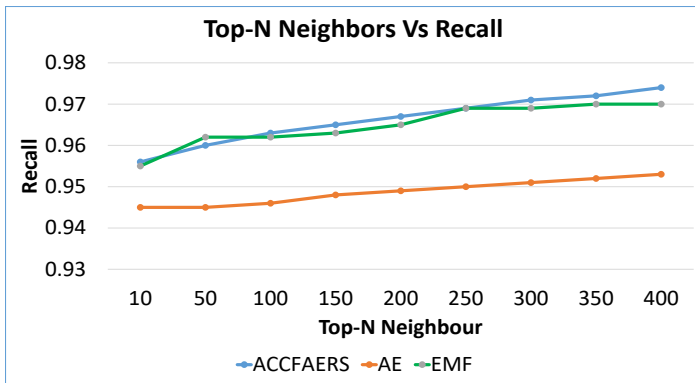


Fig. 12. Top-N Neighbor vs. Recall For Proposed method.

B. TOP-K Recommendation vs. ARHR

Results have been evaluated in this paper using the top-K recommendation versus the average reciprocal hit rank that are utilized on the collaborative filtering recommendation system. In Fig. 15, the result of average reciprocal hit rank have been shown. The more accurate the average reciprocal hit rank result in more accurate list of recommendations.

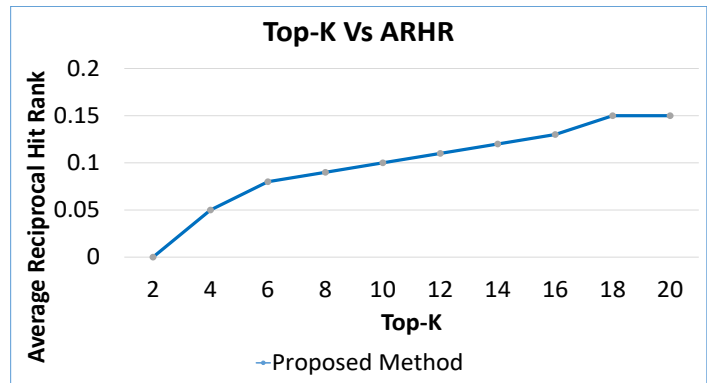


Fig. 15. Top-k Recommendation vs. ARHR.

Finally, Fig. 11, 12, 13, and 14 are used to show that recall and f-measures also increase with the increase of the neighbor size.

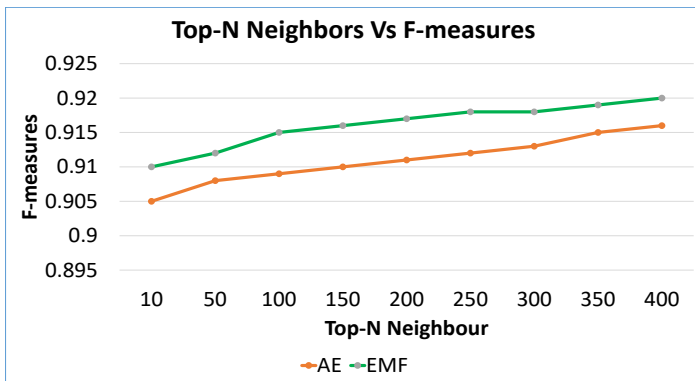


Fig. 13. Top-N Neighbor vs. F-measures For AE and EMF.

It is observed that the comprehensive approach of both autoencoder and extended matrix factorization provide better recommendation and also yield better results in terms of evaluation measures.

VII. CONCLUSION

Deep learning based methods can be called the soul of the recommender system. Performance of the recommender system mostly depends on how much accurate these methods perform. Autoencoder and matrix factorization both performs well in recommender system. They provide good performance individually but the performance of the combination of both methods is better. An approach has also been discussed in this research that how similarity between users can be found using autoencoder and extended matrix factorization for better prediction. Deep learning methods has gone further with the newer tools and techniques. Other methods like convolutional neural network, recurrent neural network, adversarial networks, and the attention based model can be used to improve the recommendation performance.



REFERENCES

- [1] R. Katarya and O. P. Verma, "Effective collaborative movie recommender system using asymmetric user similarity and matrix factorization," in *2016 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 71–75, IEEE, 2016.
- [2] N. Mittal, R. Nayak, M. C. Govil, and K. C. Jain, "Recommender system framework using clustering and collaborative filtering," in *2010 3rd International Conference on Emerging Trends in Engineering and Technology*, pp. 555–558, IEEE, 2010.
- [3] F. Roy, S. M. Sarwar, and M. Hasan, "User similarity computation for collaborative filtering using dynamic implicit trust," in *International Conference on Analysis of Images, Social Networks and Texts*, pp. 224–235, Springer, 2015.
- [4] A. Pujahari and V. Padmanabhan, "Group recommender systems: Combining user-user and item-item collaborative filtering techniques," in *2015 International Conference on Information Technology (ICIT)*, pp. 148–152, IEEE, 2015.
- [5] A. Pal, P. Parhi, and M. Aggarwal, "An improved content based collaborative filtering algorithm for movie recommendations," in *2017 Tenth International Conference on Contemporary Computing (IC3)*, pp. 1–3, IEEE, 2017.
- [6] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, p. 13, 2016.
- [7] J. Lund and Y.-K. Ng, "Movie recommendations using the deep learning approach," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 47–54, IEEE, 2018.
- [8] B. Yi, X. Shen, Z. Zhang, J. Shu, and H. Liu, "Expanded autoencoder recommendation framework and its application in movie recommendation," in *2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, pp. 298–303, IEEE, 2016.
- [9] Q. Lu, D. Yang, T. Chen, W. Zhang, and Y. Yu, "Informative household recommendation with feature-based matrix factorization," in *proceedings of the 2nd Challenge on Context-Aware Movie Recommendation*, pp. 15–22, ACM, 2011.
- [10] D. Li, R. Li, and S. Zhang, "A deep learning image recognition framework accelerator based parallel computing," in *Proceedings of the 2018 2nd International Conference on Deep Learning Technologies*, pp. 16–20, ACM, 2018.
- [11] P. Tzirakis, J. Zhang, and B. W. Schuller, "End-to-end speech emotion recognition using deep neural networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5089–5093, IEEE, 2018.
- [12] A. Utku, I. A. DoGru, and M. A. Akcayol, "Permission based android malware detection with multilayer perceptron," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2018.
- [13] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "Learning semantic representations using convolutional neural networks for web search," in *Proceedings of the 23rd International Conference on World Wide Web*, pp. 373–374, ACM, 2014.
- [14] T. Guo, J. Dong, H. Li, and Y. Gao, "Simple convolutional neural network on image classification," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, pp. 721–724, IEEE, 2017.
- [15] N. Chowdhury *et al.*, "A comparative analysis of feed-forward neural network & recurrent neural network to detect intrusion," in *2008 International Conference on Electrical and Computer Engineering*, pp. 488–492, IEEE, 2008.
- [16] E. Merindasari, M. R. Widyanto, and T. Basaruddin, "An overview of learning algorithms and inference techniques on restricted boltzmann machines (rbms)," in *Proceedings of the 2nd International Conference on Software Engineering and Information Management*, pp. 16–20, ACM, 2019.
- [17] K. George, "An application of pca to rank problem parts," in *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, pp. 341–345, ACM, 2012.
- [18] D. Novrilianto, H. Murfi, and A. Wibowo, "The singular value decomposition-based anchor word selection method for separable non-negative matrix factorization," in *2017 International Conference on Asian Language Processing (IALP)*, pp. 289–292, IEEE, 2017.
- [19] S. S. Peerzade, "Web service recommendation using pcc based collaborative filtering," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pp. 2920–2924, IEEE, 2017.
- [20] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, ACM, 2008.
- [21] Y. Koren, "Tutorial on recent progress in collaborative filtering," in *RecSys '08 Proceedings of the 2008 ACM conference on Recommender systems*, pp. 333–334, ACM, 2008.
- [22] A. Kyriakopoulou and T. Kalamoukis, "Using clustering to enhance text classification," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 805–806, ACM, 2007.
- [23] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pp. 281–286, ACM, 2006.
- [24] F. Gullo and A. Tagarelli, "Uncertain centroid based partitioned clustering of uncertain data," *Proceedings of the VLDB Endowment*, vol. 5, no. 7, pp. 610–621, 2012.
- [25] R. Florence, B. Nogueira, and R. Marcacini, "Constrained hierarchical clustering for news events," in *Proceedings of the 21st International Database Engineering & Applications Symposium*, pp. 49–56, ACM, 2017.
- [26] G. Zhao and J. Huang, "DeepSim: deep learning code functional similarity," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 141–151, ACM, 2018.
- [27] D. Hu, X. Lu, and X. Li, "Multimodal learning via exploring deep semantic similarity," in *Proceedings of the 24th ACM international conference on Multimedia*, pp. 342–346, ACM, 2016.
- [28] M. Hasan, S. Ahmed, M. A. I. Malik, and S. Ahmed, "A comprehensive approach towards user-based collaborative filtering recommender system," in *2016 International Workshop on Computational Intelligence (IWCI)*, pp. 159–164, IEEE, 2016.
- [29] L. Mendo, "Estimation of a probability with guaranteed normalized mean absolute error," *IEEE Communications Letters*, vol. 13, no. 11, pp. 817–819, 2009.
- [30] P. Prashar and T. Choudhury, "Suicide forecast system over linear regression, decision tree, naive bayesian networks and precision recall," in *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 310–313, IEEE, 2018.
- [31] P. Lingras and C. J. Butz, "Precision and recall in rough support vector machines," in *2007 IEEE International Conference on Granular Computing (GRC 2007)*, pp. 654–654, IEEE, 2007.
- [32] J. Sepulveda and S. Velastin, "F1 score assesment of gaussian mixture background subtraction algorithms using the muhavi dataset," 2015.
- [33] M. A. Najork, H. Zaragoza, and M. J. Taylor, "Hits on the web: How does it compare?," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 471–478, ACM, 2007.
- [34] D. Xiaomeng and M. Su, "Incorporating consumer browse data: Extended item-based top-k recommendation algorithms," in *In 2013 International Conference on Information Technology and Applications*, pp. 248–253, IEEE, 2013.