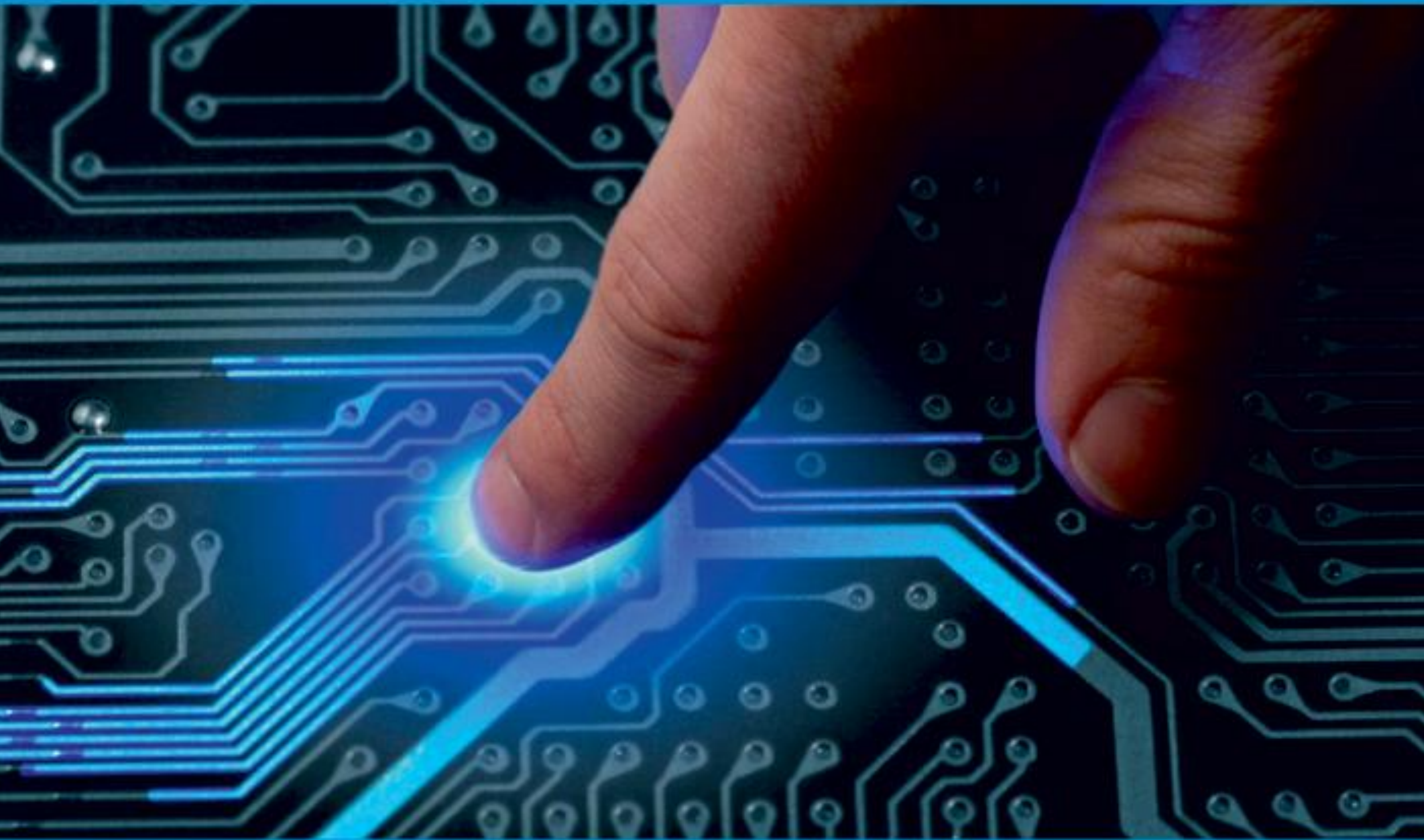
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 5, May 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

9940 572 462 | 6381 907 438 | ijircce@gmail.com | www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205354 |

Deep Learning Approach with Multiple Models for Collaborative Filtering Recommendation System

Nida Ajaj Jamadar, Dr. Deepika V. Patil, Prof. Amruta Sarudkar

PG Student, Department of AIDS, Sanjay Ghodwat University, Kolhapur, Maharashtra, India

Associate Professor, Department of CSE, Sanjay Ghodawat University, Kolhapur, Maharashtra, India

Assistant Professor, Department of Computer Engineering, Trinity College of Engineering & Research, Pune, India

ABSTRACT: Many researchers are interested in creating explicit feedback-based recommendation systems because there is a sizable quantity of explicit input, such as searching and tapping. Despite being overly challenging, explicit feedback is particularly appropriate while developing recommendation techniques. The learning potential of traditional collective filtering methods like matrix decomposition is constrained when user preferences are viewed as a linear mix of user and object latent attributes. They thus struggle with data sparsity and cold beginnings. In this project, deep neural networks will be used in addition to traditional collaborative filtering to map user and object attributes. On the other hand, scalability and data availability affect the effectiveness of the methodologies and limit the applicability of the proposals' findings. The authors then suggested combining user and item functions using the multi-model deep learning (MMDL) method to produce a hybrid RS that greatly improved. A one-dimensional neural network convolutional model which learns user and object properties is combined with a deep autoencoder in the MMDL technique to predict users' expectations. The proposed study suggests considerable success in contrast to current methods based on an in-depth analysis of three models that produce a wide range of outcomes from a single real-world dataset.

KEYWORDS: Convolutional neural networks (CNN), deep neural networks (DNN), matrix factorization, and recommender systems are a few instances of collaborative filtering.

I. INTRODUCTION

Utilizing the information retrieval method known as the Recommender System, users can receive product recommendations based on their preferences (RS). The RS is essential for addressing data congestion issues in today's internet age. With the internet's exponential growth and the growth of commercial organizations, the volume of data uploaded to the internet is increasing dramatically. The tremendous amount of online information has led to knowledge overload problems. It has been established that the RS is a practical and effective method for addressing the problems brought on by an excess of online information. Users can readily access movies, music, books, news, academic papers, and general commerce via RS. The RS has been significantly changed by websites like Google, Amazon, Netflix, YouTube, and others [7–11]. The collaborative filtering, trust-based, and content-based algorithms used by RS are only a few of the algorithms it uses. The collaborative filtering algorithm's recommendations often include people and an item engaging. No prior product or user expertise is necessary. Using a common filtering algorithm that is based on their previous response and browser history, including scores, browsing, clicks, etc., users' preferences for things can be predicted. Although a collaborative filtering algorithm is rapid and efficient, there are a few things to keep in mind, such as cold start, forecast accuracy, and the inability to record intricate user-item interactions. One of the numerous collective filtering techniques, matrix decomposition (MD), uses a vector of latent features to describe either an individual or an item, joining them into a single latent space [8, 19, 20]. The customer's interaction with an entity was then mapped using the dot product of each latent vector. As a result of the MovieLen and Netflix Awards, MD has become the accepted technique for latent model-based factor suggestion.

The MD is now being improved through a number of research initiatives, allowing for its expansion to the MD for generic function modeling [22], integration with neighborbased models [21], and combination with content models based on specific topics [22]. Despite being frequently used in collective filtering, the MD approach's effectiveness is significantly constrained by the interaction function of the dot product. Because it combines the linear multiplication of latent features, the dot product is unable to reflect the complex structure of user interface outcomes [24].

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205354 |

A number of academic fields, like text processing, speech synthesis, and image and video processing, are currently seeing excellent results from deep neural networks (DNNs). There isn't much information available on the use of DNNs in advising systems because of the wealth of literature on MD approaches. The use of current developments in DNNs' recommendation functions produces interesting outcomes. Recently, DNNs have been used in a number of trials to represent additional data, including textual object descriptions, musical aural elements, and visual information from images. When modeling recommendation systems in MD, the crucial mutual filtering effect is still exploited and expressed by fusing the latent functionality of the user and the item via an internal product. The neural network simulation technique of a shared filtering algorithm is successfully formalized in our thesis. Our attention is on the nonverbal input that consumers convey when they make decisions like buying something, uploading a movie, or clicking on something like Implicit feedback allows for automatic tracking, which is easier for service providers to obtain than explicit feedback (like reviews and ratings). Even though there aren't many issues, using it can be difficult because client loyalty isn't measured or rated.

In this work, noisy implicit feedback signals are projected using DNNs to address the research issues mentioned above. By merging the Deep Auto-Encoder Neural Network (DeepAEC) and One-Dimensional Traditional Neural Network (1D-CNN) approaches, we present a multi-model deep learning (MMDL) method that successfully boosts the collaborative filtering algorithm efficiency in our study. We thoroughly evaluated three models that produced a range of results on a single realworld dataset in order to show how well DeepAEC and 1DCNN operate as collective filtering strategies. This essay is divided into the following main sections. While Section 3 provides the method, Section 2 describes the literature review. The results of the test are available in Section 4.

II. LITERATURE SURVEY

In this section, we review significant recent works and divide them into paragraphs according to their subjects. The aforementioned assumptions have been supported by a number of model-based recommendation strategies, incorporating regression-based algorithms [15], clustering approaches [34], latent semantic approaches [33], and matrix factoring approaches [35]. The MD is the most used type of collaborative filtering. The latent properties of the person and the object are represented by vectors of the same dimension created by this technique for both users and objects. This approach is covered in the chapters on probabilistic parametric principal component analysis [38], non-singular value decomposition [36], singular value decomposition [36], and probabilistic matrix factorization [37]. For the sparse rating matrix in particular, the latent vectors learnt by MD algorithms are ineffective.

A depth MD model was created by Xue et al. [40]. Using the conventional MD method, the feature matrix of people and things is broken down. Using a multilayer feed-forward neural network, related properties are fully mined. The suggested ranking is produced by the inner product of the related lowdimensional functions. To increase the recommendation's accuracy,

The implicit feedback of SVD++ and the autoreduction encoder helped to uncover the characteristics of video data that are used in Zhang et alAuto .'s SVD++ model[41]. Collaborative filtering based on auto-encoders was created by Ouyang et al (ACF). The user score value of the item is split into five vectors by the ACF system. The method's disadvantage is that it makes By tackling the integer scoring estimation problem, the ACF algorithm loses predictive power and makes the scoring matrix sparser. Shain et al. [43] are the authors of AutoRec. Replicating the outcomes of the first input is the main objective of the AutoRec conceptThe AutoRec model corrects the problem with non-integer scoring values for predictions, but it does so without introducing additional data noise that would degrade model performance and make it more prone to overfitting.

Rankings are predicted using CDAE, which was developed by Wu et al. [44] The model receives implicit input from the user regarding the objects. Depending on the user's desire for the item's interest, each perceptron in the model's input segment that is specifically associated to an object is assigned a value of 0 or 1.

The model then sequentially makes suggestions to the user regarding the objects that are correlated with the anticipated values of the output layer convolution layers. To illustrate the results of the final forecast using material data and a scoring matrix, Strub et al. [46] suggested a CFN model. The model's suggestion accuracy has improved when compared to older methods. According to Yan et al. [45], this technique has the drawback that there is a dearth of data and the information is presented in a straightforward manner.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205354 |

Convolutional neural networks are the most widely used neural networks for computer vision and visual recognition (CNNs). Convolutional (CL), pooling (PL), and fully connected (PL) layers are components of CNN (FL). Compared to MLPs, CNNs have the same number of perceptrons and fewer parameters, making them simpler to train [47].

The CL creates n function maps, where n is the number of filters to be employed, to extract features from the input. To fix the problems brought on by the function mappings' high dimensionality curse, the PL must lower the dimensionality of features. The ConvMF[48] addresses the issue of data sparsity and improves prediction accuracy.

uses contextual data from records by integrating CNN with PMF. The bag of words method fails to address issues with scarcity because it disregards the order of the terms, degrading the text's semantic meaning. To address this problem, the final forecast report produced by the CNN model and the epsilon variable in the PMF model are combined to construct the latent document vector

III. PROPOSED METHODOLOGIES

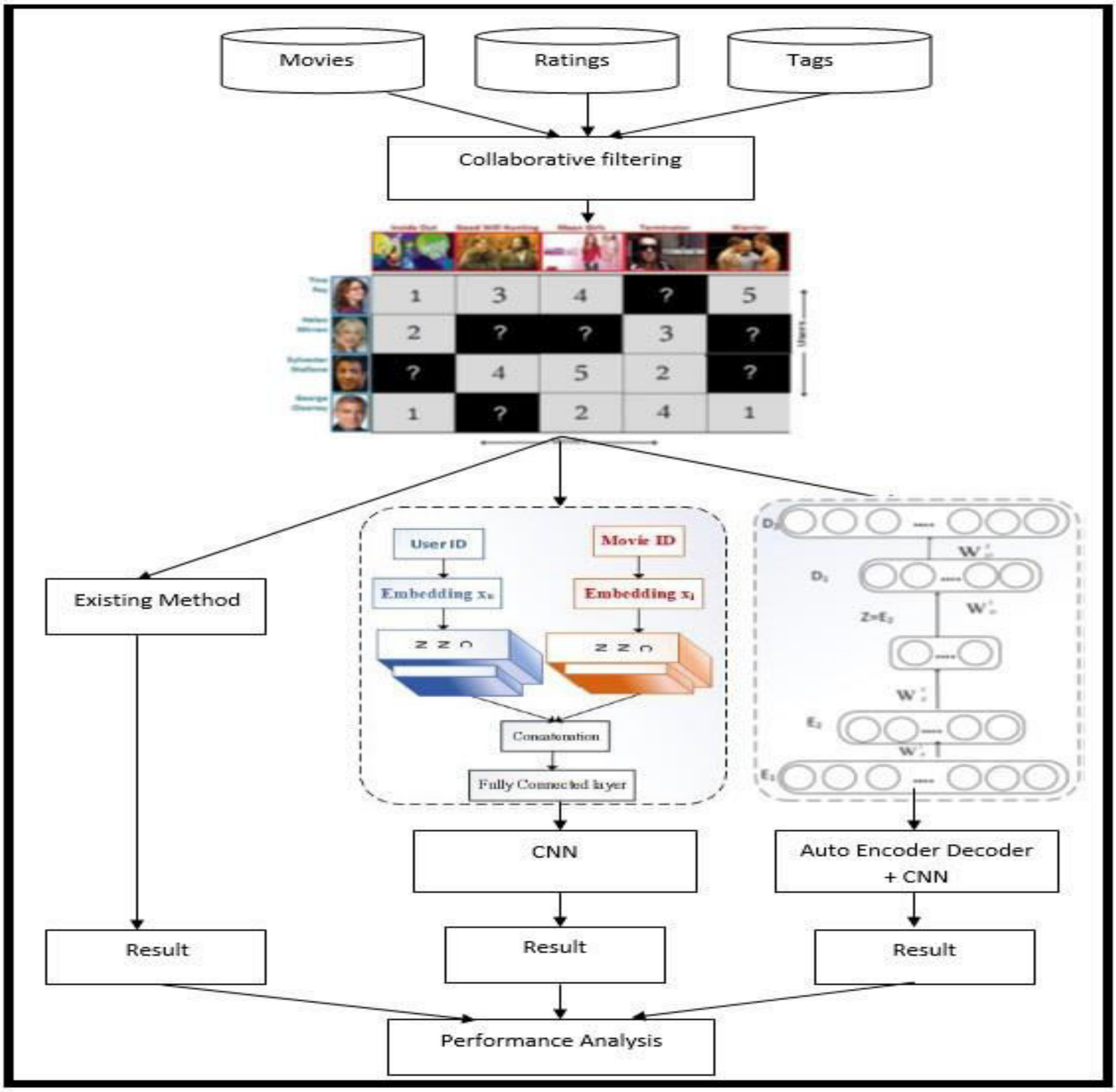


Fig. 1: MMDL Proposed Structure

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |
|| Volume 12, Issue 5, May 2024 ||
| DOI: 10.15680/IJIRCCE.2024.1205354 |

This section gives a general overview of how the framework is used. This study's goal is to investigate a collaborative filtering strategy based on explicit feedback, so as features, userID and movieID have been selected (itemID). As depicted in Fig. 1, we suggested an MMDL technique that teaches user-item interaction by combining a 1D-CNN result with a deep autoencoder (DeepACE) result. Reviews for each model are different. We start by identifying the problem based on explicit data in Section 3.1, which describes the collaborative filtering method. Section 3.2: Current Technique (Recommender Net) inserted Section 3.3 3.4 of Section uses the DeepACE model. In Section 3.5, Segment 3.5, the 1DCNN is covered. Finally, a description of the suggested MMDL model is given.

3.1 Collaborative Filtering

The recommendation task in this study is focused on the collaborative filtering algorithm's explicit feedback. In the case of implicit feedback recommendation, it is typically the case that the positive feedback is being discussed rather than the negative input, in contrast to explicit feedback, which comprises both positive and negative feedback. On a scale of 1 to 5, the degree of inclination is displayed, with "very like" representing the highest score (see Fig. 2b). Implicit feedback can only be provided by observed (chosen) and unobserved (unselected) occurrences (see Fig. 2a). The unselected state cannot be stated to have a positive tendency; Since unselected things include both things that users are actually not interested in and things that users do not find but are interested, it can only be said that they have a negative tendency.

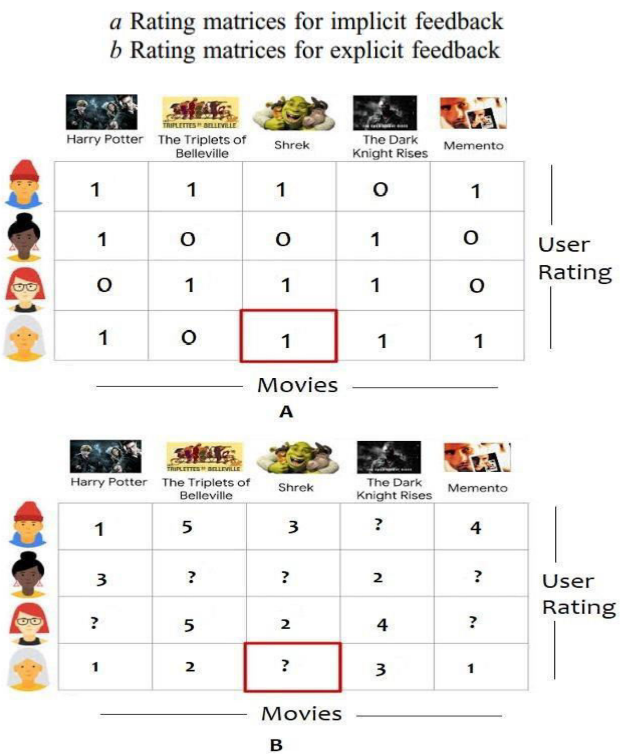


Figure: Rating Matrix -2 Simple demonstration of implicit and explicit feedback differences observed

As a result, training recommendation systems might be difficult when there are no negative parameters. Assume that m and n, respectively, stand for a group of users and objects (movies) (see Fig. 2a). The implicit feedback matrix R [Rmn for users and movies is defined as

$$r_{ui} = \begin{cases} 1, & \text{if interaction (user } u\text{, movie } i\text{) is observed;} \\ 0, & \text{otherwise} \end{cases}$$

The entry rui displays the user's rating for the film. When set to 1, rui displays a user-movie interaction. The user and the movie are not engaging if there is no communication. There is a chance for noise signals because these interactions

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205354 |

do not reveal if the user actually likes or dislikes the movie. The user's features are represented by the vector xu of that user, and the dimension of features is indicated by the letter n. The user's latent feature matrix was also set to be x [Rma. This is similar to how x [Rnb is used to denote the latent quality of movies We generate projected recommendations from explicit data in order to predict the ratings given to the ideas in R by unobserved component scores.

Five users who rated various items on a scale of 1 to 5 are represented in the matrix. For instance, the third item received a rating of 3 from the first user.

The matrix's cells are typically empty because users typically only rate a small number of objects. Every user rating or commenting on every item is extremely unlikely. The opposite of a matrix that has primarily filled cells is referred to as sparse, and vice versa for a matrix that has mostly empty cells.

How to Measure Ratings

The processes taken to establish the rating R a user U would give item I are as follows:

Utilizing the user ratings found in the previous phase to determine the rating R
Finding additional raters of things who are comparable to you.

Before you can determine the rating R that user U would give to a certain item I, you must create a list of users who are similar to user U. Similarly, to similarity, you can do this in a number of different ways.

The R rating a user assign to an item the ratings that the top 5 or top 10 people who are most comparable to you gave me are probably quite close to the average. The average rating provided by n users is represented by the equation below

$$R_U = (\sum_{u=1}^{n} R_u)/n$$

This equation states that the sum of the ratings supplied by the n users who are the most similar to you divided by the n users who are the most similar to you equals the average rating provided by the n users.

3.2 Existing Method (Recommender Net)

This dataset consists of a large number of files that include data on the users, the movies, and the ratings those users have given the films they have seen. The following ones are the ones that need attention: • u.item: the movie list
• u.data: the user-submitted rating list

The ratings are stored in a user data file that contains the user ID, item ID, rating, and timestamp. These are the first few lines of the document:

The embedding layer is then delivered to the fully linked DeepACE layers after the input layer, which use encoder and decoder functions to map the latent vectors and forecast the scores. The high-dimensional original data is supposed to be converted into a low-dimensional space using the encoding model's fully connected layers. Fully lined decoder layers are envisioned as the inverse process of the encoder network that is required to first encode and then decode the original data before it is finally fused on layer z. This conceptually resembles fully-connected encoder layers. (see (3)) utilized initial five rows of data as was already said, the movies hold the rating that a user gave a certain movie. This file contains these 100,000 ratings, which will be used to predict audience reactions to unwatched movies.

Splitting dataset into train-test: -

After reading the dataset, we must divide it into a test dataset and a training dataset. 90% of the dataset in this instance is thought of as a test, and the remaining 100% is a training dataset. For greater accuracy, neural network training needs a lot of data.

International Journal of Innovative Research in Computer and Communication Engineering
| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |
|| Volume 12, Issue 5, May 2024 ||
| DOI: 10.15680/IJIRCCE.2024.1205354 |

Train Model: -

user_id	item_id	rating	timestamp
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
244	51	2	880606923
166	346	1	886397596

After the model has been constructed, the following step is to train it using the training dataset. If you want the model to be more accurate, you can use more epochs than the five we used in this case. The embedding vector needs to be modified so that the anticipated value is as close as feasible to the actual value. The loss in this case is the difference between the actual and predicted scores over the whole training dataset.

Make Predictions: -

We can now generate predicted values for the test dataset, i.e., the predicted rating will be generated using the built-in model with respect to the user and movie identifiers. In this part, we have predicted the values of the user id and movie id in the first 10 rows from the testing data. In general, we are capable of predicting values for the full test dataset and then propose films based on the highest predicted rating for a specific user.

3.3 Autoencoder for deep neural networks

Two input vectors, xu, and xi, which, respectively, reflect the properties of userID u and movieID I are contained in the initial input layer for the DeepAEC model [51, 52]. These sparse binary vectors have been one-hot encoded. These vectors are integrated as features based on the ensuing equation.

x = Concatenate (xu, xi)

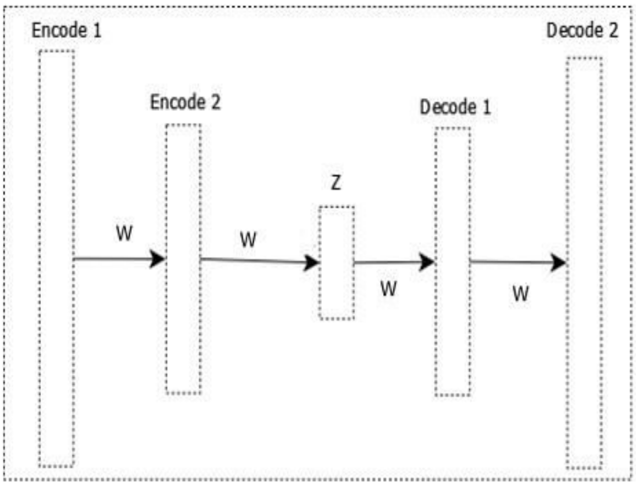


Fig. 3. Model of the encoder and decoder

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |
|| Volume 12, Issue 5, May 2024 ||
| DOI: 10.15680/IJIRCCE.2024.1205354 |

$$\left.\begin{aligned} E_1 &= \mathrm{Relu}(W_E^1)\cdot x + b_1 \\ E_2 &= \mathrm{Relu}(W_E^2)\cdot E_1 + b_2 \end{aligned}\right\}(\text{Encoder})$$
$$z = E_2$$
$$\left.\begin{aligned} D_1 &= \mathrm{Relu}(W_D^1)\cdot z + b_3 \\ D_2 &= \mathrm{Relu}(W_D^2)\cdot D_1 + b_4 \end{aligned}\right\}(\text{Decoder})$$

Where W and b represent the weight matrix and biases for each layer, and E1, E2... EL, D1, D2... DL stands for the encoder and decoder layer outputs, respectively, which are activated by the activity of the rectified linear unit (ReLU). Since relu is the most efficient, straightforward to compute, and translated activation mechanism, we choose it [53].

Relu(e) = the limit (e, 0)

Finally, at the output layer, the features acquired from the previous inner layer got projected into the output layer to produce the estimated score based on the following equation:

r^ui = (wT LDL + bL)

where d is the sigmoid function, which is given as (z) = (1/(1 + −z)).

3.4 1D convolution neural network architecture

The two vectors used as input in the 1D Classification model are UserID xu and MovieID xi [50]. Utilizing different feature extraction methods for each vector, the 1D CNN concatenates the receptive vector xu and xi into a new, smaller vector. Consider the following equation, which illustrates how a longer vector Z with size n and weight W could be changed into a small vector Y having n m + 1 element.

$$y_i = \sum_{j=m-1}^{0} z_{i+j} * w_j$$

.

where $i = (1, n - m + 1)$

A vector with length z equal to the averages of all of the values in the input data will be produced by convolution. if we have a vector of length n and the weight matrix is also of length n wi = 1/n. It is this twisted convolution that it is. A moving average of length 2 is produced if the same weight matrix is one shorter than the input matrix (based on the input of the 1D-CNN model, It has two lengthy input vectors called pu and qi, which, respectively, describe the characteristics of users and movies. the following equations demonstrate the use of PLs in the convolution layer:

xu = pooling(Conv(cuser)) xi = pooling(Conv(citem))

The pooling () function comes in two different forms: avgpool () and maxpool (). In our study, we used the MaxPooling approach, which delivers fundamental translations independent of internal representation and requires fewer learning parameters. The findings from both vectors are combined into a single long vector, which is displayed as an FL in the output layer of Fig. 4.

$$x_L(\boldsymbol{x}_u, \boldsymbol{x}_i) = \begin{pmatrix} \boldsymbol{x}_u \\ \boldsymbol{x}_i \end{pmatrix}$$
$$x_{ui}^{\mathrm{1Dcnn}} = \delta(w_L^{\mathrm{T}} x_L + b_L)$$

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205354 |

3.5 Combine the versions from 1D-CNN and DeepACE

This section illustrates complex user-item interactions using the suggested multi-modal neural network (MMDL). Combining DeepACE with 1D-CNN models is the simplest way to improve reinforcement learning because the mixture models may infer complicated user-item interactions from the data. The methods DeepACE and 1D-CNN can be used with embedding layers in a variety of ways [24]. To improve the outcomes and calculate the evaluation score of the ith customer on the ith item, we included the last hidden layers of DeepACE and 1D-CNN.

3.5.1 Training model: The creation and study of models help achieve a variety of objectives. The most popular objective functions for training recommendation systems are pair-wise, point-wise, and list-wise. the interests of users who are examining product pairs that are deemed eligible for selecting the top-N suggestions are taken into account by the pair-wise objective function. For rating prediction tasks, precise ratings are produced using the point-wise objective function. User preferences for a list of items have a significant impact on the list-wise goal functions of deep learning algorithms.

3.5.2 Recommendation making: We were able to determine a user's performance score for the movies they had never seen before after training the recommended algorithm (rated). While making suggestions for just one individual, we were able to choose the movies with said highest expected ratings.

IV. EXPERIMENTAL STUDY

In this section, we present our Section No 4.1 set-up for an Experimental, Section No 4.2 dataset description given, and Section No 4.3. Metrics for evaluation are displayed. in Section No 4.4 The settings of the Proposed Structure for MMDL experimental results are given, in Section No 4.5. finally, Performance Analysis is demonstrated.

4.1 set-up for an Experimental

Using the Windows operating system, four 3.10 GHz Intel Core i5-2400 CPUs, a 500 GB hard disc, and our model experiments in practise. We used Python 2.7 and Keras 2.0 with TensorFlow 3.0 as the backend.

4.2 Dataset Description

The Movielens rating system dataset was developed as a result of ongoing Movielens research. The publisher is GroupLens Study at the University of Minnesota. One of the most used datasets for evaluating collaborative filtering techniques is this one. This dataset is available at http://www.movielens.com in a variety of formats. To use the MovieLens 100k datasets, the performance of the suggested model as well as other models used for comparisons is assessed. The MovieLens 100k dataset contains 100,000 reviews of films. Each user has provided more than 20 ratings, with the lowest rating being 1 and the highest being 5. Because these userprovided ratings are explicit, we particularly chose this dataset to study how explicit feedback may be learned from implicit ratings. By changing each element to a 1 or a 0, which indicates whether the user has selected (ranked) the item or not, we were able to turn it into explicit data. The predicted rating is returned by all test processes, which accept (user, item) pair input parameters. After the data has been read in, the scored data is inserted in the rating matrix's user row and item column, creating matrices that are (943 1682), (6040 3952), and (100k) for MovieLens, respectively.

4.3. Metrics for evaluation are displayed.

The suggested model's prediction ability is assessed using the root means square error (RMSE) [55]. The definition is given by the equation that follows in the
RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{ui}(r_{ui} - \hat{r}_{ui})^2}$$

The genuine rating is denoted by the letters r, the overall number of predicted films is denoted by the letters n, and the predicted value for user u for a particular film is denoted by the letters rui.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205354 |

4.4 The settings of the proposed structure for MMDL experimental results

4.4.1 Existing Method (Recommender Net),

4.4.2 Neural network architecture of 1D convolution, 4.4.3 Deep neural network autoencoder these are the experimental results given in below

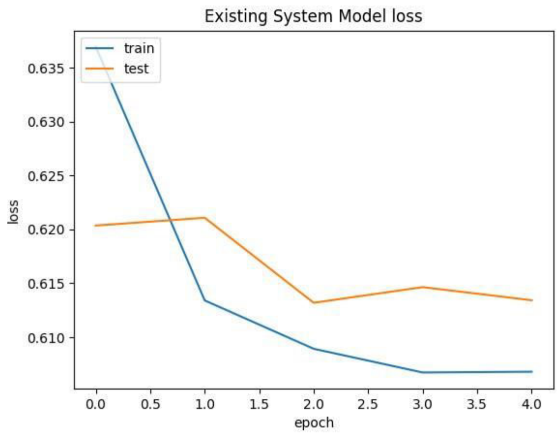
4.4.1 Existing Method (Recommender Net) num_users, num_movies, training set, and test set after the Output of the top 10 recommendations shown
in the Existing Method (Recommender Net)

Existing_top10

```
Existing System
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Recommendations Based on user  331  Review
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Recommended Top 10 movies
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 >  Star Wars: Episode V - The Empire Strikes Back (1980) : Action|Adventure|Sci-Fi
2 >  Princess Bride, The (1987) : Action|Adventure|Comedy|Fantasy|Romance
3 >  Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981) : Action|Adventure
4 >  Lawrence of Arabia (1962) : Adventure|Drama|War
5 >  Apocalypse Now (1979) : Action|Drama|War
6 >  Star Wars: Episode VI - Return of the Jedi (1983) : Action|Adventure|Sci-Fi
7 >  Chinatown (1974) : Crime|Film-Noir|Mystery|Thriller
8 >  Shining, The (1980) : Horror
9 >  Back to the Future (1985) : Adventure|Comedy|Sci-Fi
10 >  Cool Hand Luke (1967) : Drama
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Existing_plotgraph



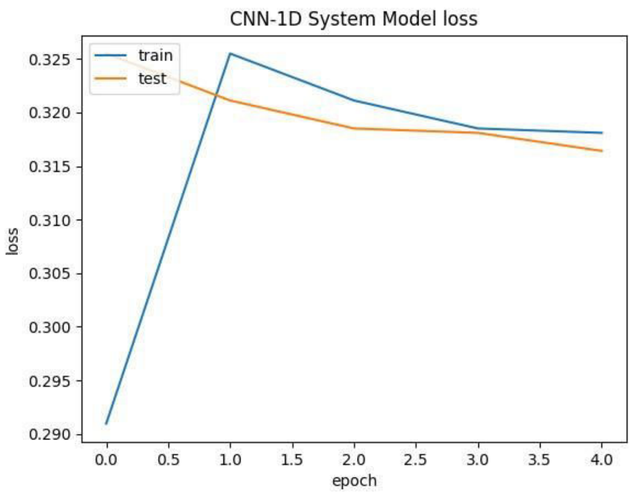
4.4.2 Neural network architecture of 1D convolution A convolutional neural network sometimes referred to as a CNN or ConvNet, is a subclass of neural networks that is particularly adept at processing data with a grid-like architecture, such as user IDs, movie IDs, and ratings, before producing an output concatenated into a fully connected network.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |
|| Volume 12, Issue 5, May 2024 ||
| DOI: 10.15680/IJIRCCE.2024.1205354 |

CNN1D_top10

```
CNN 1D Model Based
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Recommendations Based on user  167  Review
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 >  Desperado (1995) : Action|Romance|Western
2 >  Doom Generation, The (1995) : Comedy|Crime|Drama
3 >  Hard Target (1993) : Action|Adventure|Crime|Thriller
4 >  Thirty-Two Short Films About Glenn Gould (1993) : Drama|Musical
5 >  Gay Divorcee, The (1934) : Comedy|Musical|Romance
6 >  Farewell to Arms, A (1932) : Romance|War
7 >  Aladdin and the King of Thieves (1996) : Animation|Children|Comedy|Fantasy|Musical|Romance
8 >  Shall We Dance (1937) : Comedy|Musical|Romance
9 >  Saving Santa (2013) : Animation|Children|Comedy
10 >  Willy/Milly (1986) : Comedy|Fantasy
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

CNN1Dplot_graph



4.4.3 Deep neural network autoencoder

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

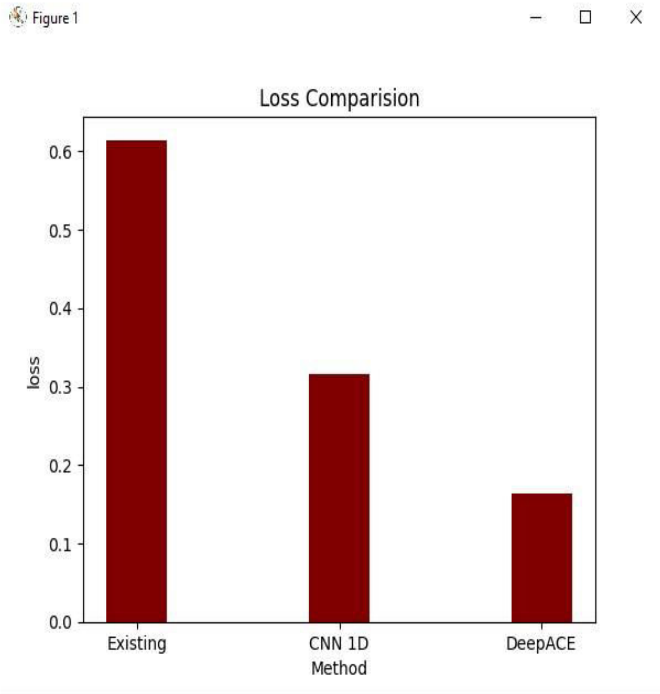
| DOI: 10.15680/IJIRCCE.2024.1205354 |

4.5 Finally, Performance Analysis Are

Demonstrated.

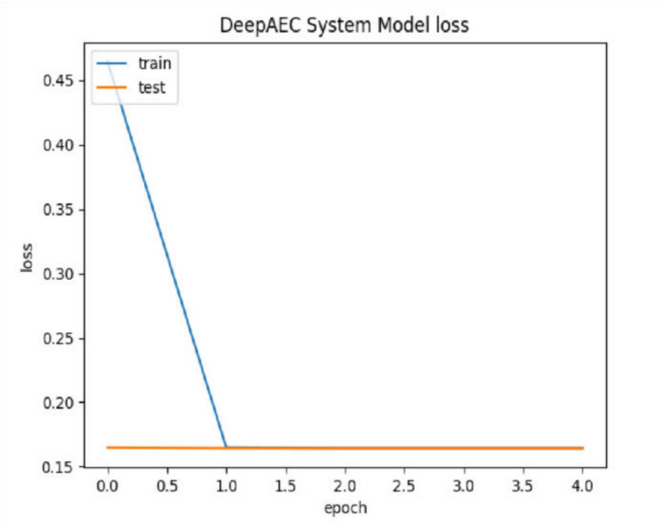
Autoencoders are self-supervised deep learning models that reproduce input data to condense it. Since they were trained as supervised deep-learning models that function as unsupervised models during inference, these models are known as self-supervised models. An autoencoder is made of two components:

1. Encoder: It serves as a compression unit and compresses the input data.
2. Decoder: It decompresses the input by reconstructing the compressed input.



AD_top10

```
DeepAEC Model Based
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Recommendations Based on user  185  Reviews
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 >  Sudden Death (1995) : Action
2 >  Moll Flanders (1996) : Drama
3 >  Underground (1995) : Comedy|Drama|War
4 >  Bloodsport 2 (a.k.a. Bloodsport II: The Next Kumite) (1996) : Action
5 >  World of Apu, The (Apur Sansar) (1959) : Drama
6 >  Mystery Science Theater 3000: The Movie (1996) : Comedy|Sci-Fi
7 >  Barbarella (1968) : Adventure|Comedy|Sci-Fi
8 >  Some Folks Call It a Sling Blade (1993) : Drama|Thriller
9 >  Run of the Country, The (1995) : Drama
10 >  He's Just Not That Into You (2009) : Comedy|Drama|Romance
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```



ADplot_graph

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205354 |

V. CONCLUSION AND FUTURE SCOPE

Collaboration filtering (CF) methods are important for organizing and developing recommendation systems. The sparsity of the data, which shows the integral nature, scalability, and matrix ratings of the data, is a disadvantage of CF approaches. In this paper, we proposed a DeepACE neural network combined with a 1D conventional neural network multi- modal deep learning approach (MMDL) for a collaborative RS. In a head-to-head comparison, the proposed model was pitted against the state-of-the-art. According to the outcomes of our experiments, the MMDL outperforms other well-known techniques in terms of RMSE measurements. The 100k MovieLens dataset was one of one real-world dataset used to assess the model. We intend to focus on explicit feedback in our upcoming work because implicit feedback is insufficient to create a complete recommendation system.

REFERENCES

[1] Carrer-Neto, W., Hernández-Alcaraz, M.L., Valencia-García, R., et al.: 'Social knowledge-based recommender system. Application to the movies domain', Expert Syst. Appl., 2012, 39, (12), pp. 10990–11000

[2] Bogdanov, D., Haro, M., Fuhrmann, F., et al.: 'Semantic audio content-based music recommendation and visualization based on user preference examples', Inf. Process. Manage., 2013, 49, (1), pp. 13–33

[3] Al-Hassan, M., Lu, H., Lu, J.: 'A semantic enhanced hybrid recommendation approach: a case study of e-government tourism service recommendation system', Decis. Support Syst., 2015, 72, pp. 97–109

[4] Kim, H.K., Oh, H.Y., Gu, J.C., et al.: 'Commenders: a recommendation procedure for online book communities', Electron. Comm. Res. Applic., 2011, 10, (5), pp. 501–509

[5] Cleger-Tamayo, S., Fernández-Luna, J.M., Huete, J.F.: 'Top-n news recommendations in digital newspapers', Knowl.-Based Syst., 2012, 27, pp. 180–189

[6] Son, J., Kim, S.B.: 'Academic paper recommender system using multilevel simultaneous citation networks', Decis. Support Syst., 2018, 105, pp. 24–33

[7] Koutrika, G.: 'Modern recommender systems: from computing matrices to thinking with neurons'. Proc. 2018 Int. Conf. on Management of Data (ACM), Houston TX, USA, May 2018, pp. 1651–1654

[8] Aljunid, M.F., Manjaiah, D.: 'Movie recommender system based on collaborative filtering using apache spark', in Balas, V., Sharma, N., Chakrabarti, A. (Eds.): 'Data management, analytics and innovation', Advances in Intelligent Systems and Computing, vol. 839 (Springer, Singapore, 2018), pp. 283–295

[9] Aljunid, M.F., Manjaiah, D.: 'A survey on recommendation systems for social media using big data analytics', Int. J. Latest Trends Eng. Technol., Spec. Issue (SACAIM 2017), 2017, pp. 48–58

[10] Nassar, N., Jafar, A., Rahhal, Y.: 'A novel deep multi-criteria collaborative filtering model for recommendation system', Knowl.-Based Syst., 2020, 187, p. 104811

[11] Aljunid, M.F., Manjaiah, D.: 'An improved ALS recommendation model based on apache spark'. Soft Computing Systems. ICSCS 2018. Communications in Computer and Information Science, Kollam, India, 2018, vol. 837, pp. 302–311

[12] Ricci, F., Rokach, L., Shapira, B.: 'Recommender systems: introduction and challenges', in Ricci, F., Rokach, L., Shapira, B. (Eds.): 'Recommender systems handbook' (Springer, Boston, MA, 2015), pp. 1–34

[13] Ekstrand, M.D., Riedl, J.T., Konstan, J.A., et al.: 'Collaborative filtering recommender systems', Found. Trends® Hum.– Comput. Interact., 2011, 4, (2), pp. 81–173

[14] O'Donovan, J., Smyth, B.: 'Trust in recommender systems'. Proc. of the 10th Int. Conf. on Intelligent user Interfaces, San Diego California, USA, October 2005, 167–

[15] Sarwar, B.M., Karypis, G., Konstan, J.A., et al.: 'Item-based collaborative filtering recommendation algorithms'. Proc. of the 10th int. Conf. on World Wide Web, Hong Kong, Hong Kong, 2001, vol. 1, pp. 285–295

[16] Zhang, H., Shen, F., Liu, W., et al.: 'Discrete collaborative filtering'. Proc. of the 39th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, Pisa, Italy, July 2016, pp. 325–334

[17] Wasid, M., Ali, R.: 'An improved recommender system based on multi-criteria clustering approach', Procedia Comput. Sci., 2018, 131, pp. 93–101

[18] Fu, M., Qu, H., Yi, Z., et al.: 'A novel deep learning-based collaborative filtering model for recommendation system', IEEE Trans. Cybern., 2018, 49, (3), pp. 1084–1096

[19] Li, S., Kawale, J., Fu, Y.: 'Deep collaborative filtering via marginalized denoising auto-encoder'. Proc. of the 24th ACM Int. on Conf. on Information and Knowledge Management, Melbourne, Australia, October 2015, pp. 811–820

[20] He, X., Zhang, H., Kan, M.Y., et al.: 'Fast matrix factorization for online recommendation with implicit feedback'. Proc. of the 39th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, Pisa, Italy, July 2016, pp. 549–558

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |
|| Volume 12, Issue 5, May 2024 ||
| DOI: 10.15680/IJIRCCE.2024.1205354 |

[21] Koren, Y.: 'Factorization meets the neighborhood: a multifaceted collaborative filtering model'. Proc. of the 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 2008, pp. 426–434
[22] Wang, H., Wang, N., Yeung, D.Y.: 'Collaborative deep learning for recommender systems'. Proc. of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Sydney NSW, Australia, August 2015, pp. 1235–1244
[23] Rendle, S.: 'Factorization machines'. 2010 IEEE Int. Conf. on Data Mining, Sydney, Australia, December 2010, pp. 995– 1000
[24] He, X., Liao, L., Zhang, H., et al.: 'Neural collaborative filtering'. Proc. of the 26th Int. Conf. on World Wide Web, Perth, Australia, April 2017, pp. 173–182
[25] Zhang, H., Yang, Y., Luan, H., et al.: 'Start from scratch: towards automatically identifying, modeling, and naming visual attributes'. Proc. of the 22nd ACM Int. Conf. on Multimedia, Orlando Florida, USA, November 2014, pp. 187–196
[26] Hong, R., Hu, Z., Liu, L., et al.: 'Understanding blooming human groups in social networks', IEEE Trans. Multimed., 2015, 17, (11), pp. 1980–1988
27] He, K., Zhang, X., Ren, S., et al.: 'Deep residual learning for image recognition'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016, pp. 770–778
[28] Collobert, R., Weston, J.: 'A unified architecture for natural language processing: deep neural networks with multitask learning'. Proc. of the 25th Int. Conf. on Machine Learning, Helsinki, Finland, 2008, pp. 160–167
[29] Van den Oord, A., Dieleman, S., Schrauwen, B.: 'Deep content-based music recommendation'. Advances in Neural Information Processing Systems, Lake Tahoe, NV, United States, 2013, pp. 2643–2651
[30] Zhang, F., Yuan, N.J., Lian, D., et al.: 'Collaborative knowledge base embedding for recommender systems'. Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 353–362
[31] Low, Y.H., Yap, W.S., Tee, Y.K.: 'Convolutional neural network-based collaborative filtering for recommendation systems'. Int. Conf. on Robot Intelligence Technology and Applications, Kuala Lumpur, Malaysia, December 2018, pp. 117–131
[32] Miyahara, K., Pazzani, M.J.: 'Collaborative filtering with the simple Bayesian classifier'. PRICAI 2000 Topics in Artificial Intelligence, Australia, 2000 (LNCS, 1886), pp. 679–689
[33] Hofmann, T., Puzicha, J.: 'Latent class models for collaborative filtering'. IJCAI'99: Proc. of the 16th int. joint Conf. on Artificial intelligence, Stockholm, Sweden, 1999, vol. 99, no. 1999
[34] Liu, J., Jiang, Y., Li, Z., et al.: 'Domain-sensitive recommendation with user-item subgroup analysis', IEEE Trans. Knowl. Data Eng., 2015, 28, (4), pp. 939–950
[35] Koren, Y., Bell, R., Volinsky, C.: 'Matrix factorization techniques for recommender systems', Computer, 2009, 42, (8), pp. 30–37
[36] Sarwar, B., Karypis, G., Konstan, J., et al.: 'Application of dimensionality reduction in recommender system-a case study', Minnesota Univ. Minneapolis Dept. of Computer Science, 2000
[37] Mnih, A., Salakhutdinov, R.R.: 'Probabilistic matrix factorization'. Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 2008, pp. 1257–1264
38] Yu, K., Zhu, S., Lafferty, J., et al.: 'Fast nonparametric matrix factorization for large-scale collaborative filtering'. Proc. of the 32nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, Boston, MA, USA, July 2009, pp. 211– 218
[39] Salakhutdinov, R., Mnih, A., Hinton, G.: 'Restricted Boltzmann machines for collaborative filtering'. Proc. of the 24th Int. Conf. on Machine Learning, Corvalis, Oregon, USA, June 2007, pp. 791–798
[40] Xue, H.J., Dai, X., Zhang, J., et al.: 'Deep matrix factorization models for recommender systems'. Proc. of the Twenty-Sixth Int. Joint Conf. on Artificial Intelligence (IJCAI-17), Melbourne, Australia, 2017, pp. 3203–3209
[41] Zhang, S., Yao, L., Xu, X.: 'AutoSVD++: an efficient hybrid collaborative filtering model via contractive auto-encoders'. Proc. of the 40th Int. ACM SIGIR Conf. on
[42] Ouyang, Y., Liu, W., Rong, W., et al.: 'Autoencoder-based collaborative filtering'. Int. Conf. on Neural Information Processing, Kuching, Malaysia, November 2014, pp. 284–291
[43] Sedhain, S., Menon, A.K., Sanner, S., et al.: 'Autorec: autoencoders meet collaborative filtering'. Proc. 24th Int. Conf. on World Wide Web, Florence, Italy, May 2015, pp. 111–112
[44] Wu, Y., DuBois, C., Zheng, A.X., et al.: 'Collaborative denoising auto-encoders for top-n recommender systems'. Proc. of the Ninth ACM Int. Conf. on Web Search and Data Mining, San Francisco, California, USA, February 2016, pp. 153–162
[45] Yan, W., Wang, D., Cao, M., et al.: 'Deep auto encoder model with convolutional text networks for video recommendation', IEEE Access, 2019, 7, pp. 40333–40346

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205354 |

[46] Strub, F., Gaudel, R., Mary, J.: ‘Hybrid recommender system based on autoencoders’. Proc. of the 1st Workshop on Deep Learning for Recommender Systems, Boston MA, USA, September 2016, pp. 11–16

[47] Alfarhood, M., Cheng, J.: ‘DeepHCF: a deep learning based hybrid collaborative filtering approach for recommendation systems’. 17th Int. Conf. on Machine Learning and Applications (ICMLA), Orlando, Florida, December 2018, pp. 89–96

[48] Kim, D., Park, C., Oh, J., et al.: ‘Convolutional matrix factorization for document context-aware recommendation’. Proc. Of the 10th AMC Conf.on Recommender System, Bosto,Massachusetta, USA, September 2016,pp.233-240

[49] KORE, MS AMRUTA, DR PROF DM THAKORE, and DRAK KADAM. "Innovation of E-commerce Fresh Agricultural Products Marketing Based on Big Internet Data Platform." (2019).

[50] Kore, A. A., D. M. Thakore, and A. K. Kadam. "Unsupervised extraction of common product attributes from E-commerce
websites by considering client suggestion." Int. J. Innov. Technol. Explor. Eng 8.11 (2019): 1199-1203.

[51] S. Saudagar, M. Kulkarni, I. Raghvani, H. Hirkani, I. Bassan and P. Hole, "ML-based Java UI for Residence Predictor," 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2023, pp. 838-843, doi: 10.1109/IDCIoT56793.2023.10053480.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE

SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details