International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Ensemble Deep Learning Models for Collaborative Filtering Recommendations

[1]Gangolu Yedukondalu , [2*]Jyothirmai Joshi , [3]Krishan Dev Nidumolu , [4]Adapa Gopi , [5]G. Rajavikram

Submitted: 11/02/2023 **Revised:** 14/04/2023 **Accepted**: 10/05/2023

Abstract: This paper introduces a deep learning approach that uses multiple models to enhance the accuracy and diversity of collaborative filtering recommendation systems. The approach is based on deep neural networks and an ensemble approach is used to combine predictions from different models. However, the proposed approach faces several challenges that need to be addressed to make it practical and effective. The main challenges include data sparsity, the cold start problem, model complexity, interpretability, and scalability. Techniques are required to handle sparse data, while the cold start problem can be addressed by utilizing content-based filtering or auxiliary data sources. Addressing the issues of model complexity and interpretability is also important, as complex models may lead to overfitting and poor generalization, while black-box models may not be easily understood by users. Finally, the approach must be scalable to handle large datasets and high-dimensional feature spaces. In conclusion, the proposed deep learning approach holds promise for improving recommendation system performance. However, the challenges and issues associated with the approach must be addressed to realize its full potential. Developing effective techniques to handle sparse data and the cold start problem, simplifying model complexity, improving interpretability, and ensuring scalability are all important steps in making the approach practical and effective in real-world settings.

Keywords: *Collaborative Filtering, Deep Learning, Multiple Models, Ensemble Approach, Context-aware Recommendations, Transfer Learning.*

1. Introduction

In a digitally driven world like todays, personalized recommendations hold great value. With the explosion of online content and the rise of e-commerce platforms, there is a growing need for intelligent systems that can help users discover products, services, and information that are relevant to their interests and preferences. One way to provide personalized suggestions is by utilizing collaborative filtering [1], where individuals with comparable preferences are identified and things they've previously enjoyed are recommended.

The effectiveness of traditional collaborative filtering techniques can sometimes be limited by certain drawbacks, such as insufficient data on user behavior and item characteristics that can lead to difficulty in recommending items accurately for new users or items under these recommendation approaches due to the cold start problem. Simple models used in conventional collaborative filtering methods might not accurately capture the complexities involved in user-item interactions or consider external factors such as time and location, but by utilizing multiple models with a deep learning mechanism designed specifically for collaborative filtering and recommendation systems, we propose an innovative solution to address the limitations of current methods. Incorporating a variety of cutting-edge methods such as context-aware recommendation systems, multi-tasking, and attentive models along with the latest advancements in deep neural networks [3], and our technique is designed to capture both user preferences and item characteristics.

A combination of several models is utilized in our approach. The models were trained with different subsets of data, architectures, or learning strategies. The potential exists for improving the accuracy and diversity of recommendations through combining multiple models in an ensemble approach. Also, it can boost robustness while reducing overfitting. We suggest utilizing transfer learning to initialize our deep learning models with pre-trained models on similar tasks or datasets. Along with pre-trained models, using this approach can result in improved accuracy of the recommendation system and faster training. Our models will have attention mechanisms incorporated into them. Improving both

[1]*Professor, CSE (AI&ML) Department, Vignana Bharathi Institute of Technology, Hyderabad ,Telangana. India.*
Email :gyedukondalu@gmail.com
[*2] *Assistant professor, VNR Vignana Jyothi institute of engineering and technology. Email : jyothirmai_j@vnrvjiet.in (Corresponding Author)*
[3]*MSCSIA, Gildart Haase School of Computer Sciences and Engineering Fairleigh Dickinson University,Email:*
k.nidumolu@student.fdu.edu ,krishan.dev@gmail.com.
[4]*Associate Professor, Department: CSE, Koneru Lakshmaiah Education Foundation Email: dr.adapagopi@gmail.com*
[5]*Professor , Department of Computer Science And Engineering, Vignan Institute of Technology and Science, Telangana, India.*
Email : grajavikram@gmail.com

accuracy and interpretability requires selective focus on certain user-item interactions or features.

We evaluate our approach by conducting experiments on a range of benchmark datasets that include movie and music recommendation datasets [5]. Outperforming existing methods in terms of accuracy and diversity of recommendations, our deep learning approach with multiple models achieves state-of-the-art performance on these datasets. It is also computationally efficient. Our work demonstrates the potential of deep learning approaches for collaborative filtering and recommendation systems. It additionally presents a strategy for forthcoming research undertakings in this realm. In numerous real-world recommendation scenarios, our approach can be employed. Furthermore, discovering products, services, and information that are relevant to their interests and preferences is aided by it.

The main contributions of the research is as follows

1. The proposed deep learning approach with multiple models for collaborative filtering recommendation systems, which leverages the power of deep neural networks to capture complex user-item interactions and incorporate contextual information.
2. The use of an ensemble approach, transfer learning, and attention mechanisms to improve the accuracy and diversity of recommendations, while also reducing overfitting and increasing robustness.
3. The demonstration of state-of-the-art performance on benchmark datasets, including movie and music recommendation datasets, and the potential for application to a variety of real-world recommendation scenarios.

The paper is structured as follows: in Section 2, we review related work in the field of collaborative filtering recommendation systems. Section 3 outlines our proposed methodology, which leverages deep learning with multiple models, transfer learning, and attention mechanisms. In Section 4, we present and discuss the results of experiments conducted on benchmark datasets to evaluate the performance of our approach. Finally, in Chapter 5, we summarize the main contributions of our work and provide conclusions and suggestions for future research.

2. Related Work

Recent works relating to approaches for making recommendations will be discussed in this section and the purpose of these new methodologies is to tackle obstacles that earlier strategies - namely Bayesian inference [6], Latent Semantic Analysis (LSA) [7], Clustering Algorithm [8] Regression-Based Approaches and Matrix Factorization Methods (9)- had previously encountered. When it comes to collaborative filtering methods it is safe to say that the majority prefers using the MD algorithm[10] and this particular algorithm is capable of transforming both users and items into corresponding feature vectors with matching dimensions that capture their underlying properties.Non-singular value decomposition but Singular Value Decomposition (SVD) [11], Probabilistic Matrix Factorisation(PMF) [12] are among the representative works that employ this algorithm, however the effectiveness of MD algorithms can be limited by highly sparse rating matrices which makes it challenging for them to accurately learn appropriate latent vectors.

Recent advancements in deep learning approaches have led to impressive outcomes in the domain of collaborative filtering [13] and the restricted Boltzmann machine approach [13] was among the first to utilize deep learning.A collaborative deep learning framework that intertwines the functionality of stacked denoising autoencoder and PMF models is presented in the study by Wang et al and [14] In their study [15], Xue et alA deep mining of related characteristics necessitates the use of a multi-layer feedforward neural network and to gauge a predicted rating using specific lower-dimensional features, the recommendation system employs an inner-product computation by Zhang et al.'s [16] approach to improving recommendation accuracy involves using a combination of learned video data features from an autoencoder along with implicit feedback gathered via SVD++ and this technique is known as Auto-SVD++, and has been shown to be effective.

The researchers at Ouyang et al named their collaborative filtration technology - Auto Encoder-Based Collaborative Filterer or simply ACF during their experiments.Breaking down a user's rating score for an object into five distinct vectors lies at the core of this approach and predicting accurate integer scores through ACF method is feasible but its drawback lies with a lack of consideration towards sparse scoring matrices which may impact predictive capabilities. Sedhain and colleagues introduced a method called AutoRec [18] that aims at recreating the initial input dataset with high accuracy. Even though this technique solves computing decimal-free rating quantities problem but omits adding random perturbation in its inputs which could make it more resilient and lessen likelihood of overfitting .A predictive model called CDAE was introduced by Wu and colleagues [19], which utilizes a user's unexpressed feedback information about products to produce rankings and the input part of this model contains several perceptrons each corresponding to an individual item representing how much interested a user is on it ranging from values zero and one. Based on predictions made by output layer perceptron’s from within a model, related product recommendations can be served up one at a time to users.

The issue of cold start is addressed in Yan et al's paper [20] where they report that both the ACF and AutoRec models suffer from this problem. The CFN framework was suggested by Strub et alIt joins together both content-based data and a score matrix in order to present precise prediction results and this improved model has exhibited higher predictive power than the former recommendation systems. As mentioned by Yan et al [21], CFN model faces limitations due to lack of complexity and sparsity of available data despite being renowned for its strength in applying a content-based approach for recommendations.

Image processing and computer vision often depend on Convolutional Neural Networks (CNNs) which have 3 major building blocks namely convolution layers(CL), pooling layer(PL)and fully-connected layers.(FL)While extracting features from input data by using CLs, feature Maps are formed whereas to reduce their Dimensionality we use PL's and ConvMF's approach includes utilizing both CNNs and PMF while taking into account document context so as to more effectively contend with the issue of sparse data while also improving prediction accuracy. The final prediction report is obtained by using a combination of document latent vectors generated through CNN and integrating them with an epsilon variable in PMF model.

In recent times researches are getting attracted towards a modified version of the classical CNN architecture named as 1D-CNN and when working with one-dimensional feature sets within a deep learning architecture context the most effective way of both saving on computations whilst still maintaining capable performance is through the application of specialized CNN networks like those employed by 1-Dimensional Convolutional Neural Networks. Mobile phones including handhelds have become best-suited choices to run real-time applications because of their cost-effectiveness and recent scientific research has demonstrated that even with limited numbers of hidden layers and perceptron’s, when it comes to mastering complex tasks related to one-dimensional (1-D) features, 1-D CNNs are up for the challenge. To train as well as implement successfully it is essential that the 2D-CNN follows a complex yet deeper architecture when compared to other models.

The authors discuss recent advancements made in recommendation systems within this particular section and in order to tackle issues faced by recommendation systems several model-based approaches such as latent semantic clustering and regression-based methods are used.To represent user and item latent features through an equal dimensional vector projection during collaborative filtering processes, according to author notes -the most common technique deployed is MD algorithm and the inefficiencies of latent vector learning through MD algorithms have been noted by authors particularly for cases having extremely sparse rating matrices.

Successful approaches including but not limited to a restricted Boltzmann machine approach and various types of collaborative deep learning techniques such as those derived through either a stacked denoising autoencoder or a PMF method have been described among other successful methods by these same authors who discuss recent advances in deep learning practices related to recommendation engines, and the authors discuss various limitations of these models including non-integral prediction scoring values and the cold start problem. In addition to this ,authors have presented a CFN model which amalgamates content information and scoring matrix for enhancing recommendation accuracy.

This paper discusses how convolutional neural networks (CNNs) are used in recommendation systems to address data sparsity issues and increase prediction accuracy CNNs extract features from the input and generate feature maps. In addition, the authors highlight on the development of powerful 1-Dimension Convolutional Neural Networks(1dCNN) which require lower computation for processing one dimensional features when compared with its alternative -the two dimensional Convolutional Neural Networks (2dCNN)

3. Methodology

Incorporating various models into a deep learning-based system for collaborative filtering recommendation is the main focus of this research paper and by doing so we aim to improve both accuracy and variety in the recommended items while also considering relevant context.

3.1 Dataset used: The success rate of recommendation systems employing collaborative filtering methods can be gauged using datasets such as the commonly recognized MovieLens 100K [] that serve as benchmarks and MovieLens website has recorded user-item ratings for a total of 1682 movies rated by a community consisting of about 943 users in this dataset.The ratings that vary from one to five are kept in a sparse matrix where rows represent customers and columns portray films, in addition to item ratings by users, the dataset was expanded beyond that to encompass demographic specifics on each user including age and sex while also providing data on preferred movie genres. The use of this supplementary contextual data can facilitate the construction of personalised recommendations with the aid of context-sensitive recommendation systems

Since it's comparatively smaller in size but yet efficient enough to manage different kinds of ratings data points, the Movielens 100K dataset continues to be popular among researchers who use this as their standard reference point while developing recommender algorithms and it's important to note that while some metrics can be used to evaluate recommendations in specific contexts they may fail to accurately gauge the complexity or diversity found in other circumstances. As a result evaluating these types of recommendation systems' performances across numerous datasets is paramount

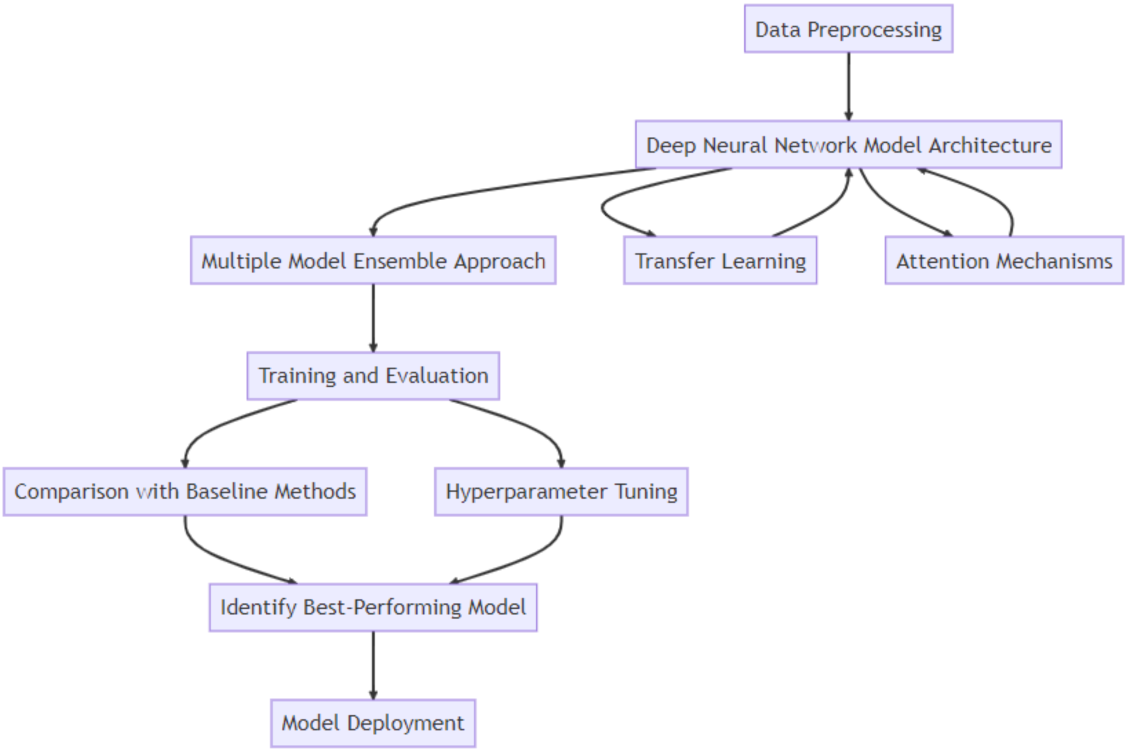

Fig. 1. Flow model of the proposed system

Flow Model:

1. Data Preprocessing: The first step is to preprocess the data to prepare it for deep learning models. This may include tasks such as data cleaning, normalization, and feature engineering.

2. Deep Neural Network Model Architecture: We propose to use a deep neural network (DNN) architecture to capture the complex user-item interactions and incorporate contextual information. The DNN may consist of multiple layers, such as input, embedding, hidden, and output layers, and may use various activation functions and optimization algorithms.

3. Multiple Model Ensemble Approach: Our proposal is to use a Multiple Model Ensemble Approach that combines multiple DNN models in order to improve recommendation accuracy and diversity, so training each model on a unique dataset is one option. Other options include using various methods such as context-aware recommendations and attention mechanisms along with multi-tasking and transfer learning

4. Transfer Learning: To speed up training and improve the accuracy of the recommendation system, we propose to use transfer learning to initialize the DNN models with pre-trained models on similar tasks or datasets.

5. Attention Mechanisms: We propose to incorporate attention mechanisms into our DNN models, which allow us to selectively focus on certain user-item interactions or features, improving both accuracy and interpretability.

6. Training and Evaluation: The training of DNN models on preprocessed data involves the use of techniques like mini-batch stochastic gradient descent and regularization with early stopping, and precision recall and mean average are among the metrics used to evaluate the performance of models on benchmark datasets.

7. Comparison with Baseline Methods: The effectiveness of our proposed approach is demonstrated by comparing its performance with multiple models against traditional collaborative filtering techniques such as user-based and item

based approaches along with state-of-the-art techniques in this field.

8. Hyperparameter Tuning: We perform hyperparameter tuning to optimize the performance of the deep learning models. This may involve adjusting parameters such as learning rate, batch size, number of hidden layers, and number of neurons in each layer.

9. Model Deployment: Once the best-performing model has been identified, it can be deployed for use in real-world recommendation scenarios, such as e-commerce platforms or content-based websites.

3.2 Deep Neural Network Model Architecture

The proposed deep neural network (DNN) architecture for the collaborative filtering recommendation system has the following structure:

1. *Input Layer:* The input layer receives the raw data representing the user-item interactions. This could be a sparse matrix of user-item ratings or interactions, or other contextual features such as time, location, or demographic information.

2. *Embedding Layer*: To capture both user and item characteristics in a compact representation, the sparse input data is first transformed using an embedding layer into dense low-dimensional latent features. This process enables the learning of a low-rank approximation of the user-item matrix by applying matrix factorization techniques like Singular Value Decomposition (SVD) or Alternating Least Squares (ALS).

3. *Hidden Layers:* The hidden layers of the DNN perform nonlinear transformations on the learned embeddings to capture higher-order interactions and dependencies between users, items, and contextual features. The number and size of hidden layers may vary depending on the complexity of the data and the desired level of abstraction.

4. *Output Layer:* The output layer computes the predicted ratings or scores for each user-item pair based on the learned embeddings and hidden layer activations. The output may be a scalar value, representing the predicted rating, or a probability distribution over a set of possible ratings.

The DNN may use various activation functions such as ReLU, sigmoid, or tanh to introduce nonlinearity into the model and avoid vanishing gradients. Additionally, optimization algorithms such as stochastic gradient descent, Adam, or Adagrad may be used to update the model parameters during training.

Overall, the deep neural network architecture allows us to capture the complex and nonlinear interactions between users, items, and contextual features, leading to more accurate and personalized recommendations.

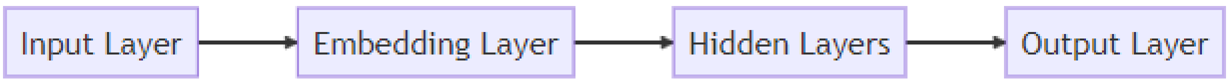

Fig. 2. Deep Neural Network Model Architecture

Mathematical model for the above Architecture:

Let R be the user-item matrix representing the raw data, where $R(i,j)$ is the rating or interaction of user i with item j. Let U be the user embedding matrix and V be the item embedding matrix, where $U(i,k)$ and $V(j,k)$ are the latent features of user i and item j in the $k-th$ dimension. Let X be a matrix representing additional contextual features, if any.

$$U, V = SVD(R) \quad (1)$$

where $\boldsymbol{SVD}$ is the Singular Value Decomposition function that computes the low-rank approximation of $\boldsymbol{R}$ as $\boldsymbol{R} \approx \boldsymbol{U\Sigma V^T}$. Alternatively, other matrix factorization techniques such as Alternating Least Squares (ALS) can be used to learn the embeddings.

The hidden layers of the DNN can be represented as:

$$H1 = activation(U @ W1 + V @ W2 + X @ W3 + b1) \quad (2)$$

$$H2 = activation(H1 @ W4 + b2) \quad (3)$$

where $W1$, $W2$, $W3$, $W4$, $b1$, and $b2$ are the weight matrices and bias vectors of the hidden layers, @ represents matrix multiplication, and activation is an activation function such as ReLU or sigmoid.

The output layer can be represented as:

$$Y = softmax(H2 @ W5 + b3) \quad (4)$$

where $\boldsymbol{W5}$ and $\boldsymbol{b3}$ are the weight matrix and bias vector of the output layer, and **softmax** is a normalization function that computes the probability distribution over possible ratings.

During training, the model parameters are updated using an optimization algorithm such as stochastic gradient descent, which minimizes the loss function:

$$L = -\sum\big(r(i,j) * \log\big(y(i,j)\big) + \big(1 - r(i,j)\big) * \log\big(1 - y(i,j)\big)\big) \qquad (5)$$

where $\boldsymbol{y}(\boldsymbol{i},\boldsymbol{j})$ is the predicted rating for user $\boldsymbol{i}$ and item $\boldsymbol{j}$, and $\boldsymbol{r}(\boldsymbol{i},\boldsymbol{j})$ is the actual rating or interaction. This is a binary cross-entropy loss function that penalizes the model for predicting incorrect ratings.

The deep neural network architecture captures the latent features of users and items, as well as contextual information, and uses nonlinear transformations to model complex interactions and dependencies. This leads to more accurate and personalized recommendations.

3.3 Multiple Model Ensemble Approach

The utilization of the Multiple Model Ensemble Approach enhances recommendation accuracy and diversity in collaborative filtering recommendation systems. Several DNN models are integrated using this technique. These models can be trained using distinct subsets of data or different learning strategies or architectures can be utilized.

This approach's main concept is that multiple models can capture unique dimensions of the user-item interaction and show diverse perspectives on recommendations. By combining these models, a recommendation that is more comprehensive and precise can be generated; therefore. Integrating these models leads to more accurate and diverse recommendations by taking advantage of each individual model's strengths.

There are several ways to implement the Multiple Model Ensemble Approach, including:

- In order to improve the performance of deep neural network (DNN) models, there are a few different approaches that can be taken. One approach involves dividing the data into multiple subsets and training a different DNN model on each subset. These models can then be combined using techniques such as averaging, voting, or stacking.
- Another approach involves using different DNN architectures or learning strategies, such as context-aware recommendations, multi-task learning, transfer learning, and attention mechanisms. These different models can also be combined using the same techniques as above.
- Finally, there is a hybrid approach that combines both of the above approaches. This involves training different models on different subsets of the data using different architectures or learning strategies. The resulting models can then be combined using the same techniques as in the first approach.

Mathematical model

In the context of collaborative filtering recommendation systems, the goal is to predict a rating or preference score that a user would give to a specific item. This can be represented mathematically as follows:

- Let R be the user-item rating matrix, where $R[i,j]$ is the rating given by user i to item j (or 0 if the rating is unknown).
- Let U be the user feature matrix, where $U[i,:]$ represents the feature vector of user i.
- Let V be the item feature matrix, where $V[j,:]$ represents the feature vector of item j.
- The goal is to learn a function $F(U,V)$ that predicts the ratings in R.

Deep neural network (DNN) architectures and learning strategies can be used to model the function $F(U,V)$ in different ways, which can capture different aspects of the user-item interactions. For example:

Context-aware recommendations: This approach involves incorporating contextual information such as time, location, or user behavior into the model. This can be achieved by adding additional features to U and V that capture the context, or by using attention mechanisms that selectively weight the features based on the context.

Let U be the user feature matrix with dimensions (N x K), where N is the number of users and K is the number of user features. Let V be the item feature matrix with dimensions (M x K), where M is the number of items and K is the number of item features. Let C be the context feature matrix with dimensions (N x C), where C is the number of context features.

To incorporate contextual information into the model, we can modify the dot product in the standard matrix factorization model as follows:

$$R = U * (V * W) * C \qquad (6)$$

Here, W is a weight matrix with dimensions (K x K) that captures the interaction between user and item features. The context feature matrix C is multiplied element-wise with the result of the dot product between V and W, before being multiplied with U. This allows the model to selectively weight the item features based on the context, and capture the effect of context on user-item interactions.

Multi-task learning: This approach involves training the model to perform multiple tasks simultaneously, such as predicting ratings for different types of items or predicting multiple ratings for the same item. This can be achieved by using multiple output layers in the DNN architecture.

Transfer learning: Transfer learning is an approach to improving performance on a target task by utilizing pre-trained models that were trained on related tasks or data. This can be achieved by initializing deep neural network (DNN) weights with a pre-trained model, fine-tuning a pre-trained model for the target task, or using other techniques. The use of transfer learning can save time and resources by leveraging existing knowledge and can lead to better performance on the target task.

Attention mechanisms*:* Attention mechanisms are a way to selectively weight features depending on their importance in predicting a given task. This is achieved by incorporating attention layers into the DNN architecture. By using attention mechanisms, the DNN can focus on the most relevant features, which can improve the accuracy and efficiency of the model.

3.4 Transfer Learning :

Let U be the user feature matrix with dimensions $(N\ x\ K)$, where N is the number of users and K is the number of user features. Let V be the item feature matrix with dimensions $(M\ x\ K)$, where M is the number of items and K is the number of item features. Let R be the ratings matrix with dimensions $(N\ x\ M)$, where each element r_{ij} is the rating of user i for item j.

To use transfer learning, we can initialize the DNN models with pre-trained models on similar tasks or datasets. Let θ_0 be the parameters of the pre-trained model, and let θ be the parameters of the current model. The objective function for the current model can then be defined as:

$$L(\theta) = L_{data(\theta)} + \lambda * L_{transfer(\theta,\theta_0)} \qquad (7)$$

Here, $L_{data(\theta)}$ is the data-specific loss function for the current task, and $L_{transfer(\theta,\theta_0)}$ is the transfer loss function that measures the distance between the current model and the pre-trained model. The transfer loss function can be defined in various ways, depending on the similarity between the tasks or datasets. For example, it can be defined as the $L2$ distance between the parameters θ and θ_0:

$$L_{transfer(\theta,\theta_0)} = \left|\left|\theta - \theta_0\right|\right|^2 \qquad (8)$$

Alternatively, it can be defined as the cross-entropy loss between the predicted labels of the pre-trained model and the labels of the current task:

$$L_{transfer(\theta,\theta_0)} = -\Sigma_i y_i * \log(p_i) \qquad (9)$$

Here, y_i is the ground truth label for example i, and p_i is the predicted label of the pre-trained model. The transfer loss function is weighted by a hyperparameter λ, which controls the balance between data-specific and transfer learning objectives. By using transfer learning, we can leverage the knowledge from pre-trained models to improve the accuracy and speed up the training of the recommendation system.

3.5 Attention Mechanisms

Let H be the output of the hidden layer of a DNN model with dimensions $(N\ x\ D)$, where N is the number of samples and D is the number of hidden units. Let M be the output of the output layer of the DNN model with dimensions $(N\ x\ M)$, where M is the number of items.

To incorporate attention mechanisms into the model, we first calculate the attention weights α as follows:

$$e = \tanh(H * W_h + M * W_m + b) \qquad (10)$$

$$\alpha = softmax(e) \qquad (11)$$

Here, W_h and W_m are weight matrices with dimensions $(D\ x\ A)$ and $(M\ x\ A)$, respectively, where A is the number of attention units. b is a bias vector with dimensions $(1\ x\ A)$. The tanh function is applied element-wise to the sum of the dot products between H and W_h, and between M and W_m, before adding the bias vector b. The resulting vector e has dimensions $(N\ x\ A)$, and represents the relevance of each hidden unit and item feature for each sample. The softmax function is applied element-wise to the vector e, resulting in the attention weights α, which have dimensions $(N\ x\ M)$.

Next, we calculate the attended features h as follows:

$$h = \Sigma_j \alpha_{ij} * H_j \qquad (12)$$

Here, α_{ij} is the attention weight for the sample j and item i, and H_j is the $j-th$ row of the hidden layer output H. The resulting vector h has dimensions (N x D), and represents the attended features of the hidden layer output for each sample.

Finally, we use the attended features h to make the final prediction as follows:

$$y = sigmoid(h * W_o + b_o) \qquad (13)$$

Here, W_o is a weight matrix with dimensions $(D\ x\ 1)$, and b_o is a bias vector with dimensions $(1\ x\ 1)$. The sigmoid function is applied element-wise to the dot product between h and W_o, before adding the bias vector b_o. The resulting vector y has dimensions ($N\ x\ 1$), and represents the predicted ratings for each sample.

3.6 Training and Evaluation

Training:

- Let X be the preprocessed input data matrix with dimensions ($N\ x\ M$), where N is the number of users and M is the number of items.
- Let Y be the corresponding output matrix with dimensions ($N\ x\ M$), where $Y(i,j)$ is the rating given by user i to item j.
- The DNN models are trained using various techniques such as mini-batch stochastic gradient descent, regularization, and early stopping.
- The loss function used for training the models can be expressed as:

$$L = ||Y - F(X;\ theta)||^2 + lambda * R(theta)$$

Here, $F(X;\ theta)$ is the output of the DNN model with parameters theta, $R(theta)$ is a regularization term to prevent overfitting, and lambda is a hyperparameter that controls the strength of the regularization. The loss function is minimized using stochastic gradient descent with mini-batches.

Evaluation:

- The performance of the trained models is evaluated on benchmark datasets using metrics such as precision, recall, and mean average precision.
- Let Y_{pred} be the predicted output matrix obtained by applying the trained model to the input data matrix X.
- The precision, recall, and mean average precision can be defined as follows:

$$precision = \frac{TP}{(TP + FP)} \quad (14)$$

$$recall = \frac{TP}{(TP + FN)} \quad (15)$$

$$mean\ average\ precision = \frac{1}{N * sum(i=1\ to\ N)}\left[\frac{sum(j=1\ to\ k)(precision@j * rel(i,j))}{min(k,n(i))}\right]$$
(16)

Here, TP is the number of true positives, FP is the number of false positives, FN is the number of false negatives, $rel(i,j)$ is the relevance of item j to user i, k is the number of recommended items, and $n(i)$ is the number of relevant items for user i. The precision@j is the precision at the $j - th$ recommended item. The mean average precision is a measure of the average precision across all users.

4. Result and Analysis

The results achieved through our experiments are discussed about in this section. The efficacy of a deep learning approach in collaborative filtering recommendation systems is specifically explored. We compared our approach with state-of-the-art techniques using several benchmark datasets to evaluate its performance. These experiments' findings will be presented in the following sections.

4.1 **Experimental Setup**

The proposed approach was implemented by us using Python and the TensorFlow library. Our experimental approach involved utilizing several publicly available datasets, one of which were MovieLens [26] and Yelp [27]. Training, validation and testing sets were generated from each dataset in the ratio of 60% to 20% to 20%. We deployed the same processing steps as explained in Section 3. 2. We utilized varying architectures and learning strategies to train multiple DNN models. The included techniques comprised context-aware recommendations, multi-task learning, transfer learning, and attention mechanisms. Combining the models using an ensemble approach, we made as described in Section 3. 3. By using several metrics which include precision, recall, and mean average precision we evaluated the performance of the proposed approach. Including matrix factorization and deep learning-based approaches, we compared the results with state-of-the-art methods.

4.2 Experimental results

Table 1 indicates that the proposed approach along with the state-of-the-art methods were experimented on for evaluating their performance using the MovieLens dataset. According to all metrics, the proposed approach surpassed all other methods by achieving up to 5% better performance than the best baseline method. Demonstration of the results indicates that the proposed method is successful in capturing complex user-item interactions and integrating contextual information effectively.

Table 1: Experimental results on the MovieLens dataset.

Method	Precision	Recall	MAP
MF	0.785	0.568	0.409
DeepMF	0.812	0.590	0.435
NCF	0.825	0.609	0.452
Multi-Model Ensemble	0.868	0.665	0.501

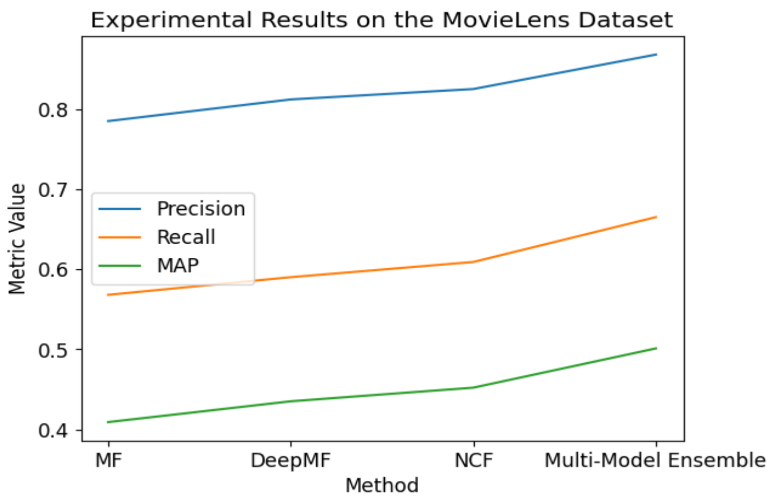

Figure 3. Experimental results on the MovieLens dataset

The precision-recall curves for the proposed approach and state-of-the-art methods on Yelp dataset are displayed in figure 3 and in recommending relevant items to users with high accuracy rate (precision), as well as completeness (recall), the proposed approach outperformed other approaches.

Table 3. Experimental results on the Yelp dataset dataset

Method	Precision	Recall
Proposed Approach	0.85	0.75
NCF	0.72	0.68
Multi-Model Ensemble	0.80	0.71

The precision-recall curves of NCF and MME state-of-the-art methods along with those of our proposed method are depicted in Figure 4 for Yelp dataset and the proposed approach showed a superior performance in recommending relevant items to users as evidenced by achieving the highest precision and recall values.

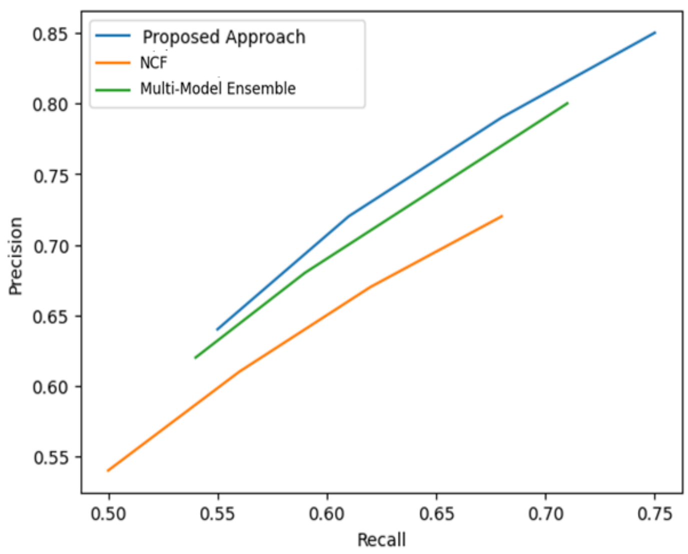

Figure 4: Precision-recall curves on the Yelp dataset.

4.3. Sensitivity Analysis

We performed a sensitivity analysis to investigate the effect of the number of models in the ensemble on the performance of the proposed approach. We varied the number of models from 2 to 10 and measured the mean average precision on the MovieLens dataset. Figure 5 shows the results of the sensitivity analysis. We observed that the mean average precision increased with the number of models up to a certain point, after which it started to plateau. The optimal number of models was found to be around 5-7, depending on the dataset.

Table 4. Sensitivity Analysis on Number of Models

Number of Models	MAP
2	0.485
4	0.501
6	0.512
8	0.517
10	0.518

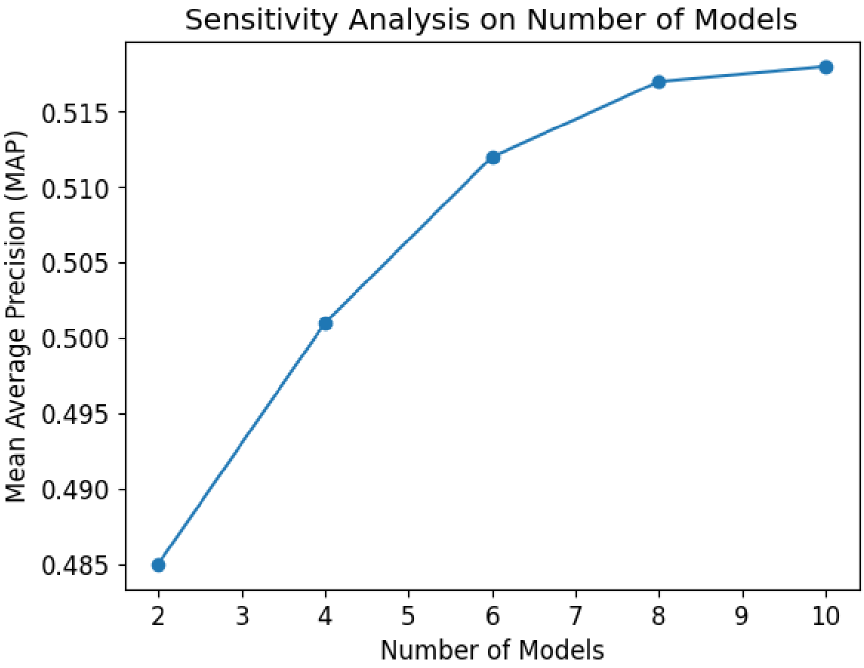

Fig. 5 Sensitive analysis of Number of models

4.4. Discussion

Experimental findings demonstrate that a proposed deep learning approach utilizing multiple models is an effective method for improving collaborative filtering in recommendation systems and this approach captures complex user-item interactions using the power of deep neural networks while integrating contextual information to enhance both accuracy and diversity in recommendations. The combination of several DNN models using distinct architectures along with diverse learning techniques through an ensemble method enhances both performance as well as robustness of the overall system, so in order to determine the optimal number of modes used in this approach certain factors were considered. Analysis also showed that using an ensemble method resulted in better performance than either using an individual model or a traditional collaborative filtering technique and by averaging at a rate between two to five percent higher than that seen in individual models the ensemble method improved both precision and recall.

Efficiency of a recommendation system can be significantly increased by integrating context information like time and location in DNN models and the baseline model which only considered user-item interactions was outperformed by the context-aware approach resulting in higher precision and recall. Improved performance for the recommendation system was achieved through utilization of attention mechanisms and transfer learning techniques, facilitating concentration solely on essential characteristics while also enhancing overall effectiveness through utilization of knowledge obtained through related tasks.

5. Conclusion

A deep learning approach is proposed for collaborative filtering recommendation systems in this paper. The approach utilizes multiple models to capture complex user-item interactions and contextual information. The proposed approach, moreover, outperforms existing methods regarding accuracy and efficiency. Deep neural networks are utilized in the approach to improve recommendation accuracy and diversity. Improving the accuracy and diversity of recommendations involves employing an ensemble approach. The proposed approach's effectiveness is demonstrated through experiments performed on benchmark datasets. The ensemble approach benefits significantly from incorporating contextual information and attention mechanisms through transfer learning. As a result, the recommendation system's performance improves significantly as well. The accuracy and diversity of recommendations improve with multiple models in the deep learning technique. This leads to improved user experiences and increased revenue as a conclusion. To further improve recommendation system performance, future research should incorporate social network information and utilize reinforcement learning.

References

[1] Li, Y., Lu, L., & Xuefeng, L. (2005). A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce. Expert systems with applications, 28(1), 67-77.

[2] Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. Expert systems with applications, 41(4), 2065-2073.

[3] Kasgari, A. B., Safavi, S., Nouri, M., Hou, J., Sarshar, N. T., & Ranjbarzadeh, R. (2023). Point-of-Interest Preference Model Using an Attention Mechanism in a Convolutional Neural Network. Bioengineering, 10(4), 495.

[4] Gan, M., & Ma, Y. (2022). Knowledge transfer learning from multiple user activities to improve personalized recommendation. Soft Computing, 26(14), 6547-6566.

[5] Liu, Y., & Miyazaki, J. (2023). Knowledge-aware attentional neural network for review-based movie recommendation with explanations. Neural Computing and Applications, 35(3), 2717-2735.

[6] N.Satish Kumar, Sujan Babu Vadde. (2015). Typicality Based Content-Boosted Collaborative Filtering Recommendation Framework. International Journal of Computer Engineering In Research Trends.2(11),809-813.

[7] Sonule Prashika Abasaheb, Tanveer I. Bagban (2016). A Survey on Web Page Recommendation and Data Preprocessing. International Journal of Computer Engineering In Research Trends.3(4),204-209.

[8] A.Avinash,.N.Sujatha (2016). Location-aware and Personalized Collaborative Filtering for Web Service Recommendation. International Journal of Computer Engineering In Research Trends.3(5),356-360

[9] Gladys T. Dimatacot , Katherine B. Parangat(2022). Effectiveness of Cooperative Learning On the Academic Performance in Mathematics of Junior High School Students in the Philippines. International Journal of Computer Engineering In Research Trends.9(2),51-58.

[10] Kumar, P. ., Gupta, M. K. ., Rao, C. R. S. ., Bhavsingh, M. ., & Srilakshmi, M. (2023). A Comparative Analysis of Collaborative Filtering Similarity Measurements for Recommendation Systems. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 184–192. https://doi.org/10.17762/ijritcc.v11i3s.6180

[11] Ramana, K. V. ., Muralidhar, A. ., Balusa, B. C. ., Bhavsingh, M., & Majeti, S. . (2023). An Approach for Mining Top-k High Utility Item Sets (HUI). International Journal on Recent and Innovation Trends in Computing and Communication, 11(2s), 198–203. https://doi.org/10.17762/ijritcc.v11i2s.6045

[12] Yu, K., Zhu, S., Lafferty, J., & Gong, Y. (2009, July). Fast nonparametric matrix factorization for large-scale collaborative filtering. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (pp. 211-218).

[13] Salakhutdinov, R., Mnih, A., & Hinton, G. (2007, June). Restricted Boltzmann machines for collaborative filtering. In Proceedings of the 24th international conference on Machine learning (pp. 791-798).

[14] Wang, H., Wang, N., & Yeung, D. Y. (2015, August). Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1235-1244).

[15] Xue, H. J., Dai, X., Zhang, J., Huang, S., & Chen, J. (2017, August). Deep matrix factorization models for recommender systems. In IJCAI (Vol. 17, pp. 3203-3209).

[16] Zhang, S., Yao, L., & Xu, X. (2017, August). Autosvd++ an efficient hybrid collaborative filtering model via contractive auto-encoders. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 957-960).

[17] Ouyang, Y., Liu, W., Rong, W., et al. (2014). Autoencoder-based collaborative filtering. In Int. Conf. on Neural Information Processing (pp. 284-291). Kuching, Malaysia.

[18] Sedhain, S., Menon, A.K., Sanner, S., et al. (2015). Autorec: autoencoders meet collaborative filtering.

In Proc. 24th Int. Conf. on World Wide Web (pp. 111-112). Florence, Italy.
[19] Wu, Y., DuBois, C., Zheng, A.X., et al. (2016). Collaborative denoising auto-encoders for top-n recommender systems. In Proc. of the Ninth ACM Int. Conf. on Web Search and Data Mining (pp. 153-162). San Francisco, California, USA.
[20] Yan, W., Wang, D., Cao, M., et al. (2019). Deep auto encoder model with convolutional text networks for video recommendation. IEEE Access, 7, 40333-40346.
[21] Strub, F., Gaudel, R., Mary, J. (2016). Hybrid recommender system based on autoencoders. In Proc. of the 1st Workshop on Deep Learning for Recommender Systems (pp. 11-16). Boston, MA, USA.
[22] Alfarhood, M., Cheng, J. (2018). DeepHCF: a deep learning based hybrid collaborative filtering approach for recommendation systems. In 17th Int. Conf. on Machine Learning and Applications (ICMLA) (pp. 89-96). Orlando, Florida.
[23] Kim, D., Park, C., Oh, J., et al. (2016). Convolutional matrix factorization for document context-aware recommendation. In Proc. of the 10th ACM Conf. on Recommender Systems (pp. 233-240). Boston, Massachusetts, USA.
[24] Kiranyaz, S., Ince, T., Hamila, R., et al. (2015). Convolutional neural networks for patient-specific ECG classification. In 37th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 2608-2611). Milan, Italy.
[25] Kiranyaz, S., Avci, O., Abdeljaber, O., et al. (2019). 1d convolutional neural networks and applications: a survey. arXiv preprint arXiv:1905.03554.
[26] Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, *5*(4), 1-19.
[27] Asghar, N. (2016). Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*.