Electronic Research Archive

ERA, 33(2): 1185–1206.
DOI: 10.3934/era.2025053
Received: 17 December 2024
Revised: 13 February 2025
Accepted: 17 February 2025
Published: 03 March 2025

https://www.aimspress.com/journal/ERA

Research article

Modeling and application of implicit feedback in personalized recommender systems

Hui Li[1], Shuai Wu[1], Ronghui Wang[1], Wenbin Hu[1,*] and Haining Li[2,*]

[1] School of Computer Engineering, Jiangsu Ocean University, Jiangsu 222000, China
[2] Department of Neurology, General Hospital of Ningxia Medical University, Ningxia 750003, China

* **Correspondence:** Email: hwb1008@163.com, lhnwww@126.com; Tel: +8615396981785, +8613709599000.

Abstract: Traditional recommendation algorithms usually rely on the user's existing data and historical behavioral records to make recommendations, which often leads to low recommendation accuracy and insufficient personalized experience. To solve these problems, this paper proposes an innovative recommendation algorithm model, neural collaborative filtering with multiple attention mechanism (NCF-MAH). The goal of this model is to enhance the effectiveness of the recommender system. The specific implementation includes constructing a negative sample set and applying matrix decomposition techniques to map user and item IDs to a low-dimensional embedding vector space. In addition, the model processes these embedding vectors using a multi-head attention mechanism to transform them into query vectors, key vectors, and value vectors, and further computes the attention scores and the corresponding weighted sums. Finally, the score prediction is accomplished by fusing the output of the multi-head attention mechanism with the results of the multilayer perceptual machine. The experimental results show that the NCF-MAH model exhibits significant advantages over the baseline model in two key evaluation metrics, hit rate and normalized discount cumulative gain (NDCG), on the MOOC platform and other datasets. Specifically, hit rate and NDCG improved by 13% vs. 9.8% and 15.7% vs. 12.8% when Top-k was set to 10 and 20, respectively.

Keywords: neural collaborative filtering; multi-head attention mechanism; implicit feedback; multiple attention force mechanisms; score prediction

1. Introduction

With development over time and the advancement of technology, social platforms and search engines not only provide convenience, but also generate a large amount of information [1]. The amount of data that users face every day is exploding. The explosive growth of data on the one hand greatly enriches the user's life; on the other hand, the overly redundant data causes great interference in the user's behavioral decisions [2]. This phenomenon, commonly termed information overload, has prompted researchers to develop recommender systems as a countermeasure. By collecting and analyzing users' behavioral data—including browsing histories, click patterns, and evaluation records—these systems identify individual preferences to deliver targeted recommendations [3]. This methodology has become both a widely studied solution and an extensively implemented strategy in digital environments.

The core of recommendation algorithms is to actively provide personalized services to users based on the interaction history between users and resources [4]. Traditional recommendation algorithms include content-based recommendation algorithms and collaborative filtering recommendation algorithms [5]. With the development of artificial intelligence (AI), building recommendation algorithms based on deep learning has become the mainstream approach.

Currently, deep learning (DL)-based recommendation models show superior performance compared to most linear-based collaborative filtering (CF) methods, which mainly utilize deep neural networks (DNNs) to capture higher-order features to understand the complex relationships between users and resources. For example, Sinha and Dhanalakshmi [6] proposed neural network matrix decomposition by combining neural networks and matrix decomposition, He et al. [7] proposed neural collaborative filtering by combining neural networks and collaborative filtering, and Fu et al. [8] proposed to capture implicit user-item interactions with a feed-forward neural network, which is more accurate in acquiring co-occurring relationships between users and resources as compared to traditional processing methods. Pan et al. [9] added social relationships as auxiliary information to the idea of collaborative deep learning, and proposed to learn social representations via a sparse superposition denoising autoencoder to solve the problem of data sparsity in social networks. Feng et al. [10] combined rating-oriented probabilistic matrix factorization and a pairwise ranking-oriented Bayesian personalized ranking together to address cold-start scenarios. Bai et al. [11] proposed cold-start KT to address this problem, which guides learning from short sequences and ensures accurate predictions for longer sequences. It also introduces cone attention to better capture complex hierarchical relationships between knowledge components in cold-start scenarios. Alfarhood and Cheng [12] proposed using matrix decomposition on the information of the ratings matrices learned by multilayer perceptron machines with the information of the resource ratings learned by convolutional neural networks. Saifudin et al. [13] utilized a mixture of feedback behind users, items, and tags to recommend tags. These recommendation algorithms utilize deep learning techniques centered on learning feature vector representations of users and resources. The training phase usually uses pairwise or pointwise loss functions to optimize the network parameters, while the recommendation phase is based on the learned user and resource vectors for matching and recommendation.

Although deep learning-based recommender systems have achieved some results, there are still some problems. Some models simply equate user behavior with explicit preferences when dealing with implicit feedback data, ignoring the noise in the data. For example, in a student course

recommendation scenario, a student may choose a course for credit rather than personal interest. Existing models fail to distinguish this effectively, leading to the misjudging of users' true preferences and affecting recommendation accuracy. In matrix decomposition techniques, some models rely on basic methods, for which it is difficult to capture the complex nonlinear relationships between users and resources, limiting the in-depth understanding of interactions. In addition, many models perform poorly on the cold-start problem and cannot fully utilize the limited user or course information for accurate recommendation. Implicit feedback data has more noise, which may mislead model learning if used directly for training. Therefore, how to reduce the noise and accurately extract the real user preferences has become an important challenge for recommender systems. Meanwhile, it is difficult for traditional algorithms to comprehensively capture complex user-resource interaction patterns, and how to optimize the model structure to improve its generalization ability remains a difficult problem in recommender systems.

This study aims to address these issues by focusing on how to effectively utilize implicit feedback data, optimize the model structure to better capture the complex interactions between users and resources, and improve the model's ability to generalize across different datasets and cold-start scenarios. Therefore, a personalized recommendation algorithm model based on implicit feedback, neural collaborative filtering with multiple attention (NCF-MAH), is proposed. First, the inner products of potential features of users and resources are taken using matrix factorization. At the embedding layer, user and resource IDs are mapped into a high-dimensional embedding vector space, and the embedding vectors are mapped into query vectors, key vectors, and value vectors. Then, weighted sum vectors are computed by calculating the corresponding scores for each attention header. Finally, the output vectors are combined with the results of the multilayer perceptron processing the implicit vectors in terms of weight ratios to produce a prediction of user preferences. The main contributions are as follows:

1) We present a novel neural network architecture for user and resource modeling. Departing from prior single architectures relying solely on matrix decomposition or MLP, it innovatively integrates generalized matrix facorization (GMF) and MLP. User-item data are one-hot encoded into sparse vectors and embedded in a low-dimensional space. GMF and MLP layers then process linear and nonlinear features respectively, with results fused for prediction. This design comprehensively captures complex user-resource interactions and enhances the model's feature-handling ability.

2) We introduce a multi-attention mechanism. Differing from traditional applications, this study linearly transforms the user, resource embedding vectors, and rating information via distinct weight matrices. Each attention head can focus on user-resource-rating interactions from diverse perspectives, capturing data patterns more precisely and thus improving recommendation accuracy and personalization.

3) We employ the binary cross-entropy loss function. Unlike previous simple uses, this study optimizes it by integrating the model's overall architecture and implicit feedback data characteristics. During training, it effectively measures the gap between predicted and real values, updating model parameters iteratively through gradient descent. This enables the model to better adapt to data and structure, enhancing prediction accuracy and generalization. Experiments show the model outperforms traditional methods significantly in metrics like hit rate (HR) and normalized discounted cumulative gain (NDCG).

2. Preliminary

2.1. Explicit vs. implicit feedback

Early recommendation models primarily relied on explicit feedback, such as users' historical ratings of resources, to predict their ratings for target items. This approach was based on the idea that by estimating the ratings users might give to target resources, top-K recommendations could be made by ranking items according to these predicted scores [14]. However, the results were often unsatisfactory, as explicit feedback fails to account for negative feedback. Users may avoid rating resources they dislike, and ignoring this absence of feedback can lead to sub-optimal model performance [15]. In contrast, implicit feedback, while abundant and easy to collect, is noisy. For example, a user purchasing an item does not necessarily indicate that they like it; they may have bought it as a gift or later realized they do not like the product. Implicit feedback also introduces challenges due to the presence of negative samples, which are difficult to identify and account for. Most existing studies treat implicit feedback as mere additional input features, failing to fully explore its intrinsic value.

To overcome these limitations, this study introduces a multi-head attention mechanism designed to better process implicit feedback data. The multi-head attention mechanism allows the model to analyze implicit feedback from multiple perspectives by processing it in parallel across different subspaces. Each attention head focuses on distinct feature dimensions and interaction patterns, enabling the model to more accurately capture users' preferences and behavioral patterns. Moreover, the model distinguishes between two types of data by encoding missing values or browsing behaviors in the user-resource interaction matrix as binary signals. This approach enhances the model's ability to capture users' behaviors and intentions from various angles, improving recommendation accuracy and revealing hidden characteristics within implicit feedback data, as illustrated in the table below.

Table 1. Representative examples of explicit and implicit feedback in each website.

	Explicit feedback	Implicit feedback
Video website	User ratings for videos	Logs of users watching videos and browsing video pages
E-commerce sites	User ratings of products	Purchase log, Browse log
News site	User ratings for news	Read the journal of the news
Learning sites	User ratings for courses	Notes taken by students on the course

2.2. Multilayer perceptron

The multilayer perceptron was developed from the perceptron, and its main feature is that it has multiple neuron layers that can process nonlinear data [16]. The basic model structure includes an input layer, a hidden layer, and an output layer, where the number of hidden layers can be more or less, the input layer to the hidden layer can be regarded as a fully connected layer, and the hidden layer to the output layer can be regarded as a classifier. Ordinary recommendation algorithms apply vector multiplication for user features and resource features to predict ratings, and each user feature is multiplied with each resource feature one by one, which consumes time and occupies space [17],

as shown in the following formula:

$$F(x) = G(b^{[2]} + W^{[2]^T}(f(b^{[1]} + W^{[1]^T}x))), \tag{1}$$

where W is the weight matrix, b is the bias vector, and f is the activation function.

Therefore, the model in this paper utilizes the advantages of multilayer perceptrons for nonlinear data processing, transforms the original vector multiplication through the multilayer perceptron, inputs the user features and resource features obtained from the model into the multilayer perceptron, and the final output value is the predicted rating value, as shown in the following figure.

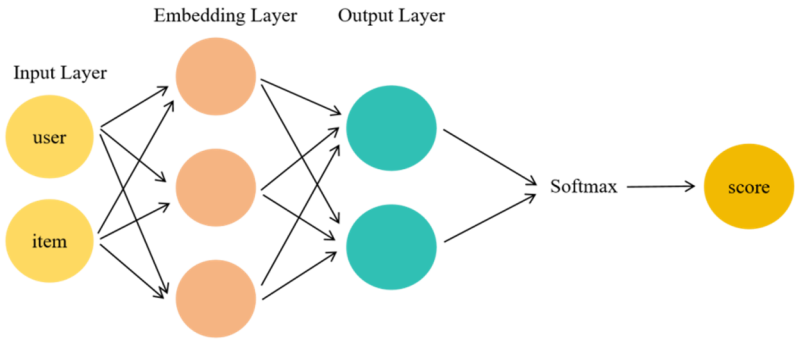

Figure 1. Schematic diagram of a multilayer perceptron.

2.3. Generalized matrix decomposition

The basic idea of matrix decomposition recommendation algorithms [18] is to decompose the user-resource rating matrix R into two low-dimensional user feature matrices U and resource feature matrices V as shown in the following Eq (2). In the process of rating prediction, the user and the resource are usually represented as a two-dimensional matrix form, i.e., the user-resource rating matrix. Koren [19], who introduced implicit feedback into recommendation systems through the SVD++ model, addressing cold-start issues caused by sparse explicit ratings. Subsequently, the temporal SVD++ algorithm extended this framework by incorporating time-sensitive mechanisms, dynamically adjusting recommendation weights through two key operations: decaying historical user behavior influence while amplifying recent neighbors' implicit feedback patterns. Different from traditional matrix decomposition models, this method deeply integrates generalized matrix factorization with a multilayer perceptron. Traditional matrix decomposition models mainly focus on mining potential features of users and items, with which it is difficult to deal with complex nonlinear interactions. In our model, on the other hand, the GMF layer is responsible for capturing linear and low-order nonlinear relationships, and the MLP layer handles high-order nonlinear relationships, which complement each other and enable the model to portray the complex user-resource interactions more comprehensively.

$$R_{m\times n} = U_{m\times d}V_{n\times d}^{T}. \tag{2}$$

Here n denotes resources quantity and d the latent feature dimensions for users/resources. Matrices U (user preferences) and V (resource attributes) model observed rating while predicting unratedinteractions. To optimize formula alignment with real-world ratingdata, the matrix factorization

algorithm employs linear regressionprinciples, constructing the following objective function:

$$J = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{i,j}(R_{i.j} - U_i V_j^T)^2 + \frac{\lambda}{2}(\|U_i\|^2 + \|V_j\|^2). \quad (3)$$

In this Eq (3), m represents the number of users, $I_{i,j}$ is an indicator parameter with a value of 1 if user i has ever rated resource j and 0 otherwise, is the actual rating of user i on resource j, is the predicted rating, and λ is a regularization parameter to prevent overfitting.

2.4. Multi-attention mechanisms

The attention mechanism [20] is one of the major breakthroughs in the field of deep learning, which has been widely used in computer vision, natural language processing, and other fields. As the process of the weighted transformation of features, the multi-head self-attention mechanism [21] is an attention mechanism in which each head is calculated in the same way, and only the parameters are different, so as to be able to represent features from multiple subspaces, and compared with the ordinary self-attention mechanism, it can obtain features in multiple dimensions. The user and resource embedding vectors are input to the attention module as query, key, and value, respectively. Since each user can be associated with multiple resources, the batch size can be larger than 1. The attention module computes the user's attention score for the resource, and the output represents an aggregated representation of the user's weighted resource embedding vector.

3. Personalized recommendation algorithm model based on implicit feedback

To address the above issues, we propose a personalized recommendation algorithm model based on implicit feedback, named NCF-MAH, as shown in Figure 2. We will now detail its components.

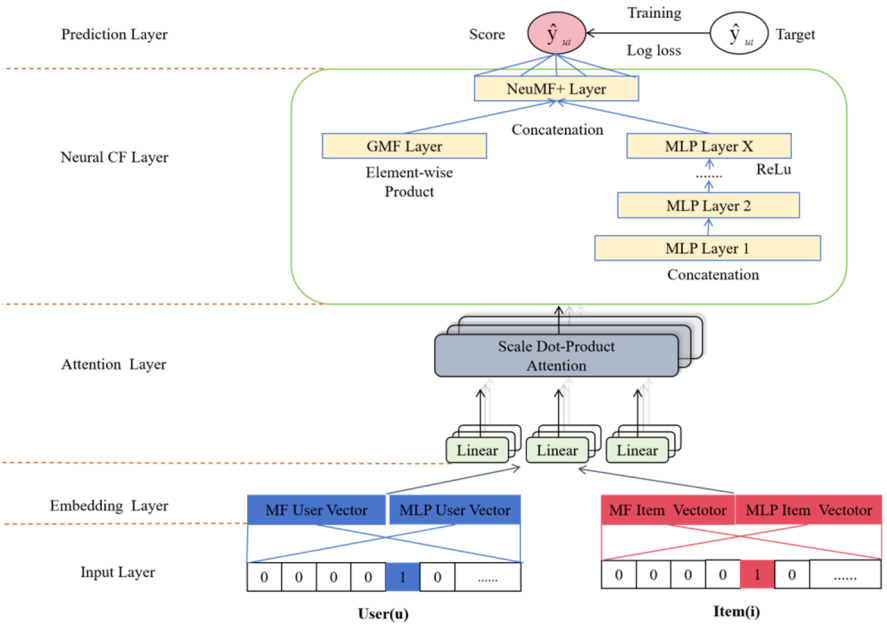

Figure 2. NCF-MAH model architecture.

3.1. Input and preprocessing layer

In the NCF-MAH model, the first type of data we need to process is the interaction information between users and resources. These data are typically presented in matrix form, where rows represent users, columns represent resources, and the element values in the matrix indicate the strength of interaction between users and resources, such as rating scores.

To effectively utilize these data in constructing a recommendation model, we convert the input User-Item data into sparse vectors through one-hot encoding, as defined below:

$$y_{ui} = \begin{cases} 1, & if\ user\ interacts\ with\ resource \\ 0, & if\ user\ does\ not\ interacts\ with\ resource \end{cases}. \tag{4}$$

Specifically, for M users and N resource items, we can transform each user and item into $1 \times M$ and $1 \times N$ vectors, respectively. For example, the vector for the i-th user is ([0, 1, 0, ..., 0]) (where the i-th element is 1 and the rest are 0), indicating that this user has interacted with the i-th resource. Similarly, the vector for the j-th item is ([0, 0, ..., 0, 1]) (where the j-th element is 1 and the rest are 0), indicating that this item has interacted with the j-th user. Afterward, we embed the user and item vectors into a lower-dimensional space, multiplying the input vector N with the embedding matrix P to obtain the embedded vector of this vector.

3.2. Enhanced attention layer

In this paper, we improve the performance of NCF models by introducing a multi-head attention mechanism. The multi-head attention mechanism is an innovation in the Transformer architecture that allows for more complex linear transformations of the output vectors of the previous layer to capture the interactions between users, resources, and ratings in more detail. The multi-head attention layer first performs independent linear transformations on the user and item embedding vectors, the user embedding matrix is U of size $m \times d_u$, and the item embedding martix is V of size $n \times d_v$, where m is the number of users, n is the number of items, d_u is the dimension of user embeddings, and d_v is the dimension of item embeddings. For each user u and item i, the initial approach to acquire their embedding vectors u_vec and i_vec is through direct lookup from pre-trained embeddding tables. Then, the multi-head attention layer first performs independent linear transformations on the user embedding vector u_vec, the item embedding vector i_vec, and their rating information. These transformations are realized by different weight matrices, denoted as W_Q, W_K, and W_V. To generate the query vector Q_u for user u, we multiply the user embedding vectors u by W_Q, i.e., $Q_u = W_Q \times u$. Similarly, for the key vector K_i of item i, we have $K_i = W_K \times i$, and for the value vector V_i, $V_i = W_V \times i$, the item matrix V contributes to the computation of the key, query, and value matrices by providing the item embeddings. Each column of the item matrix represents an item's embedding in the low-dimensional space. When computing the key, query, and value vectors for a particular user-item pair, the corresponding item embedding from the item matrix is retrieved and linearly transformed using the weight matrices W_K and W_V along with the user embedding being transformed using W_Q. Next, we compute the attention weights of user u and item i by computing the dot product of Q_u and K_i, and then apply the scaling factor and softmax function. Subsequently, we merge the vectors of weighted values of all resources to obtain a weighted representation of user u. Finally, this weighted representation is then linearly transformed once more to generate the final output vector to be used as input for the next layer, as follows:

$$Attention(Q_u, K_i, V_i) = softmax(\frac{Q_u K_i^T}{\sqrt{d_k}})V_i, \tag{5}$$

$$Output_u = W_o * WeightedSum_u, \tag{6}$$

where d_k is the dimension of the key vectors and Wo is the weight matrix of the final linear transformation, and the specific principle is shown in Figure 3.

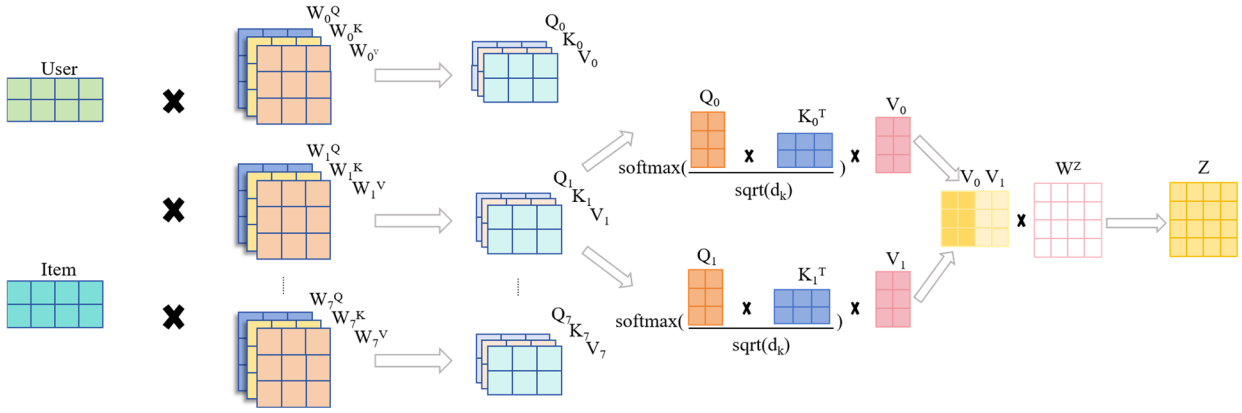

Figure 3. Schematic diagram of the attention layer.

3.3. Output and prediction layer

To enhance the non-linear fitting ability of the attention mechanism, in the GMF model, to capture the interaction between users and items in a distinct way, we focus on the embedding multiplication of the user-resource vectors p_U^G and q_I^G to obtain the embedding matrices P^G and Q^G. This element-wise multiplication operation is a fundamental step in GMF, aiming to highlight the specific relationships between user characteristics and item features encoded in the embeddings. Then, we calculate the scores through a linear layer and the sigmoid activation function, resulting in the vector ϕ_{ui}^{GMF}. In the MLP model, we concatenate the user-resource vectors p_u^M and q_i^M to obtain P^M and Q^M matrices, then calculate the scores through fully connected layers and ReLU activation functions, resulting in the vector ϕ_{ui}^{MLP}. We then connect these two vectors and pass them to the neural layer that maps the vectors to a one-dimensional space to get the final predicted score, as follows:

$$\begin{aligned} Z_{ui} &= \alpha\phi_{ui}^{GMF} + (1-\alpha)\phi_{ui}^{MLP} \\ \hat{y}_{ui} &= \sigma(Z_{ui}^T h) \end{aligned}, \tag{7}$$

where α is a hyper-parameter that controls the contribution ratio of the GMF and MLP parts, and h is the weight vector of the final neural layer. This combination allows the model to leverage the advantages of both GMF's simple interaction capture and MLP's complex pattern learning ability.

The model is a multi-layer fully connected neural network. Hidden layers use the ReLU activation function to boost non-linear expressiveness and learn complex feature relationships. The output layer applies the sigmoid function to map results to probability scores for predictions. The binary cross-entropy loss function (BCE loss) measures the gap between predicted and actual values. A gradient descent algorithm iteratively updates model parameters to enhance prediction accuracy and generalization. This iterative process enables the model to adapt to various datasets and make

more accurate predictions.

3.4. Improved model pseudo-code

Algorithm 1: NCF-MAH algorithm

Input: training data D={(*u*,*i*,*r_ui*)}, where *u* is a user, *i* is a resource, and *r_ui* is a rating, pre-trained embeddings of user *U* and resource *V*, MLP layer parameters *W_mlp*, *b_mlp*, Multihead Attention Mechanism parameters *W_at*t, *b_att*, and Fully Connected Layer parameters *W_fc*, *b_fc*

Output: trained NCF-MAH model

1.initialize model parameters θ={*U*,*V*,*W_mlp*, *b_mlp*, *W_att*, *b_att*, *W_fc*, *b_fc*}
2.for each interaction in the training set (*u*,*i*,*r_ui*)
3.*u_vec=U[u]*
4.*i_vec=V[i]*
5.*umlp*=ReLU(*W_mlp* * *u_vec*+*b_mlp*)
6.*imlp*=ReLU(*W_mlp* * *i_vec*+*b_mlp*)
7.*uatt*,*iatt*=MultiHeadAttention(*umlp*,*imlp*,*W_att*,*b_att*)
8.x=[*uatt*;*iatt*]
9.*r_ui=W_fc***x*+*b_fc*
10.*L*=LossFunction($\hat{r}_{ui}$, $\hat{r}_{ui}^{true}$)
11.Update θ to minimize *L*

4. Experiments

4.1. Dataset

To critically evaluate the proposed methodology's effectiveness, this paper experiments on two real-world datasets: MOOC [22] and EdX (https://www.kaggle.com/datasets/edx/course-study). The MOOC Cube dataset from X School has 706 MOOC courses, 38,181 videos, 114,563 interactions, and 199,199 users. The EdX dataset is based on 290 edX online courses from Harvard and MIT, with 250,000 certifications, 4.5 million participants, and 28 million hours of data. Table 2 shows detailed dataset statistics. To fairly assess comparison methods, we adopt the same data-processing as in [23]. Each instance in the training or test set is a sequence of historical lessons paired with a target lesson. In training, the last course in a sequence is the target, and the others are historical. Each positive example pairs with 1000 randomly sampled negative ones. In testing, each test-set history course is the target, and the corresponding training-set course of the same user is historical. As in [24], each positive instance pairs with 100 randomly sampled negative ones to form the test data.

Table 2. Statistics of the experimental datasets.

Dataset	#Courses	#Uers	#Interactions	#Average interactions
MOOC	698	199199	682753	3
EdX	188	289	290	1

4.2. Experimental setup

This study completed comparative experiments of the improved NCF algorithm and baseline algorithms in an environment based on Python 3.8, PyTorch 1.11.0, CUDA 11.3, and on an RTX 3090. The specific parameter settings for both datasets are shown in the following figure, where "-" indicates the same parameters as the previous column.

Table 3. Experimental hyperparameter settings.

Parameters	MOOC	EdX
topk	10, 20	-
num_factors	8	-
num_negatives	4	-
lr	0.001	-
num_heads	4	-
head_dim	8	-
test_num	1000	290
Epoch	100	50
Batch_size	2560	64
Optimizer	Adam	-
Loss function	BCELoss	-
Activation function	ReLu	-

4.3. Evaluation metrics

The following metrics are used in this paper to evaluate the performance of all models, which are widely used in other related work.

The hit rate (HR) is mainly used to measure whether the recommended list contains resources that users are really interested in. Specifically, the hit rate of the first K resources is a recall-based metric, and HR@K, which represents the percentage of resources successfully recommended to users, is defined as follows:

$$HR@K = \frac{\sum_{u=1}^{U} Hits_u@K}{|GT|}, \tag{8}$$

where *GT* refers to the set of basic facts for all users in the test set, which the number of resources in the top-K recommendation list for the *u*-th user belonging to the test set, and | | denotes the size of the set.

Normalized dicounted cumulative gain (NDCG) evaluates the ranking performance by considering the position of the correct resource. Specifically, NDCG@K for the top K resources is an accuracy-based metric that considers the predicted positions of different user recommendation lists. The specific definitions are as follows:

$$NDCG@K = \frac{1}{U}\sum_{u=1}^{U} \frac{DCG@K}{IDCG_u@K}, \tag{9}$$

$$DCG_u@K = \sum_{i=1}^{K} \frac{2^{rel_u^i}-1}{log_2(i+1)}, \tag{10}$$

$DCG_u@K$ denotes the ideal discounted cumulative revenue realized by the best top-k recommendation list for the u-th user, and rel_u^i is the hierarchical correlation between the i-th recommendation result and the u-th user.

Precision, a metric used to evaluate the performance of a recommender system, focuses on the number of correct recommendations in the recommendation results. Specifically, for the first K recommendation results, Precision@K is the ratio of the number of correctly recommended items to the total number of recommendations, calculated as follows:

$$Precision@K = \frac{1}{U}\sum_{u=1}^{U}\frac{|Rel_u \cap Rec_u(K)|}{K}, \tag{11}$$

where U is the total number of users, Rel_u denotes the set of relevant resources for the user u, and $Rec_u(K)$ denotes the set of the top-k items recommended for the user.

5. Results and analysis

5.1. Comparison experiment

To verify the effectiveness of the proposed model, the NCF-MAH model was compared with other baseline models, including the classic matrix factorization model (GMF), supervised learning model (MLP), and neural network-based collaborative filtering model (NeuMF). A brief overview of these models is as follows:

MLP [25]: It uses MLP on a pair of user and course embeddings to generate recommendation probabilities.

GMF [26]: It directly models the linear relationship between users and resources through the dot product of their low-dimensional embedding vectors, achieving personalized recommendations.

NeuMF [7]: It utilizes neural networks to merge user and resource embedding representations, learning their non-linear interactions for user recommendations.

FM [27]: The interactions between features are modeled by mapping each feature to a low-dimensional vector space and computing the inner product between these vectors, thus effectively capturing the nonlinear relationships implicit in the data.

Wide&Deep [28]: Combining the memory capabilities of generalized linear models and the learning capabilities of deep neural networks, it can simultaneously capture known feature interactions and discover new complex patterns to provide more accurate personalized recommendations.

Comparative experiments between the MLP, GMF, NeuMF models, and the NCF-MAH model were conducted using the MOOC and EdX datasets. The top-k values were set at 10 and 20, separately, and the model experimental curves are shown in the following figures, with specific values presented in the tables below.

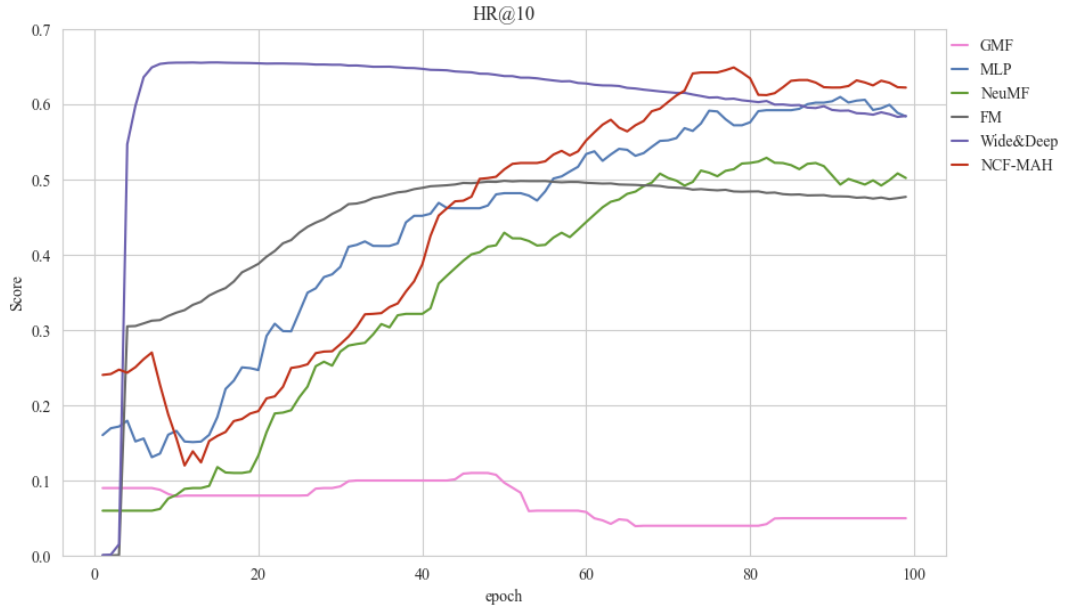

Figure 4. K = 10, Performance of different algorithmic models for HR on MOOC dataset.

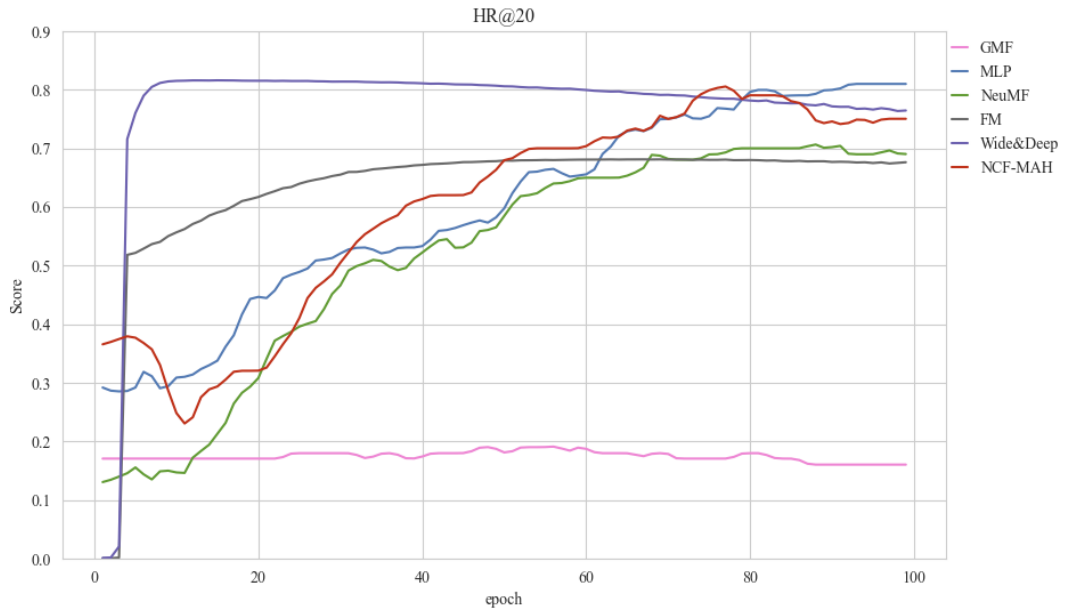

Figure 5. K = 20, Performance of different algorithmic models for HR on MOOC dataset.

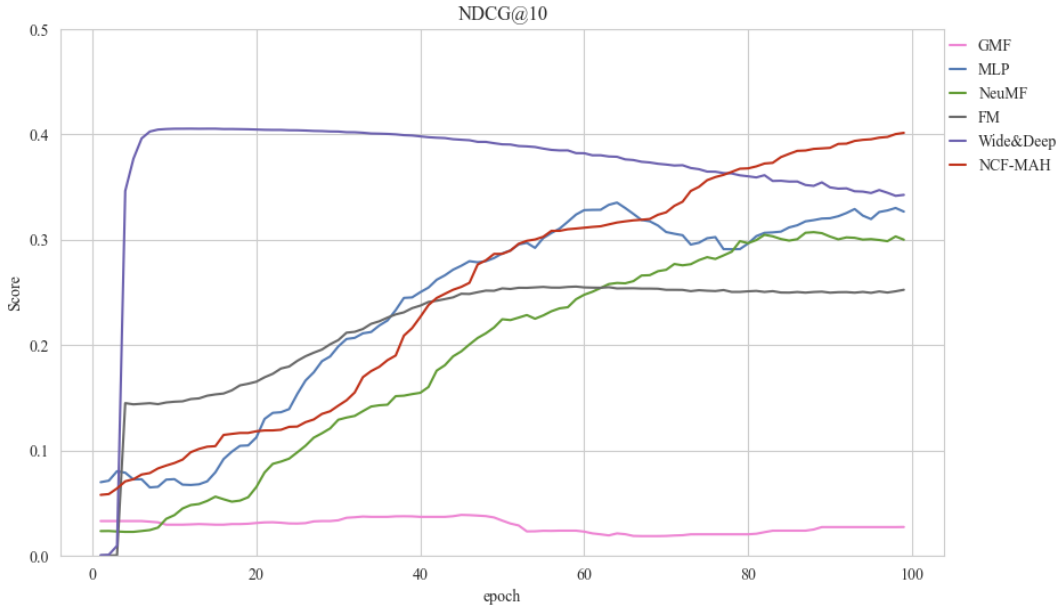

Figure 6. K = 10, Performance of different algorithmic models for NDCG on MOOC dataset.

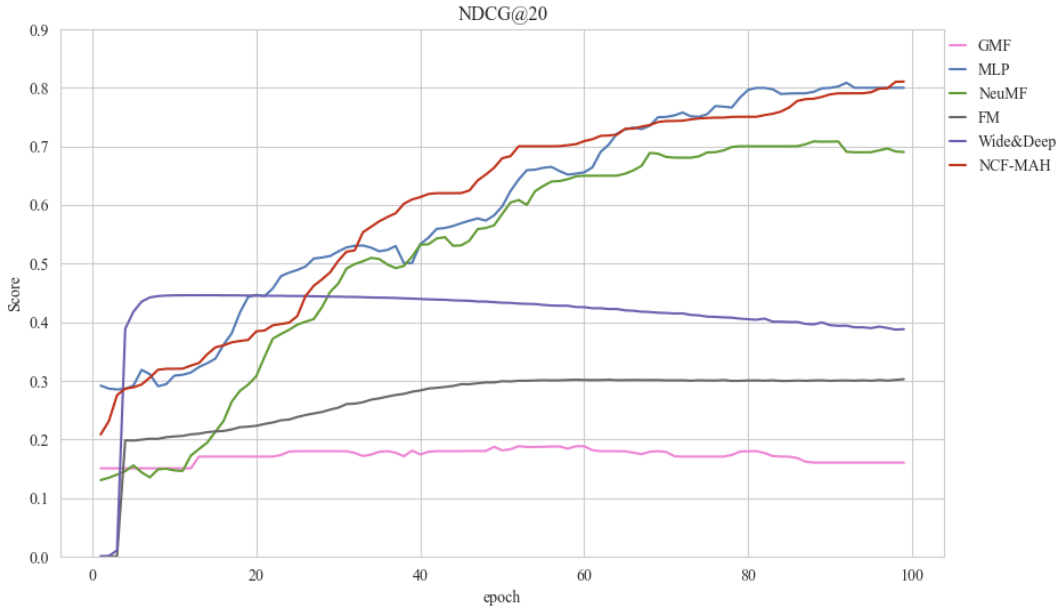

Figure 7. K = 20, Performance of different algorithmic models for NDCG on MOOC dataset.

From the experimental results, the NCF-MAH model shows higher prediction accuracy compared with MLP, GMF, and traditional NCF models on two different sized datasets. As can be seen from Figures 4–7, on the MOOC dataset containing more than 600,000 pieces of data, the improved model improves the HR@10, HR@20, NDCG@10, and NDCG@20 metrics by 13%, 9.8%, 10.3%, and 10.2%,

respectively. In contrast, as can be seen in Figures 8–11 that the small EdX dataset with about 300 data entries shows greater improvements of 15.7%, 12.8%, 7.5%, and 10.9%, respectively.

This is mainly due to the addition of the multi-head attention mechanism in the NCF model, which allows the model to simultaneously focus on multiple different information dimensions, with each dimension handled by a separate attention head. This way, different heads can learn various aspects of the user-resource interactions, enriching the model's understanding of the data. For instance, one head might focus on the user's interest in learning course categories, while another head could concentrate on the duration of the user's engagement with learning courses. Through this approach, the model gains a more comprehensive understanding of user behavior and preferences.

During this process, each head independently calculates the attention scores between user embedding vectors and item embedding vectors, and then these scores are summed up. This total score is weighted and summed with the vectors calculated using the GMF and MLP methods. This structure enables the model to better handle the complex relationships between different users and resources while maintaining personalized recommendations.

Additionally, we use the BCE loss function as the model's loss function to optimize its performance. The BCE loss effectively measures the difference between the model's predictions and the true values, allowing the model to update parameters more efficiently during training, thus improving prediction accuracy.

These results clearly indicate that the proposed NCF-MAH model demonstrates superior performance compared to traditional baseline models when handling datasets of different scales.

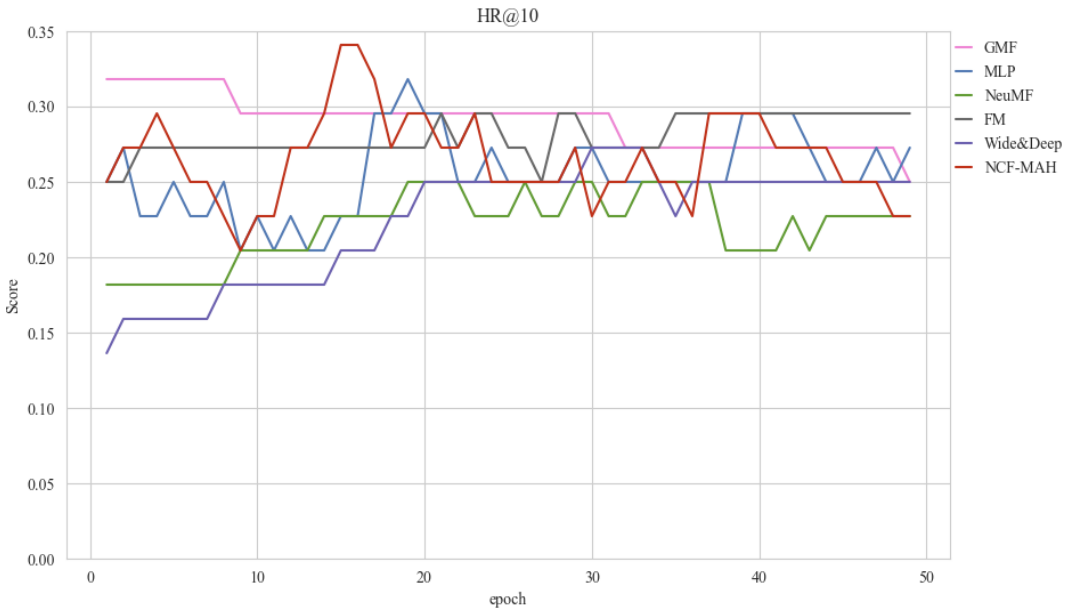

Figure 8. K = 10, Performance of different algorithmic models for HR on the EdX dataset.

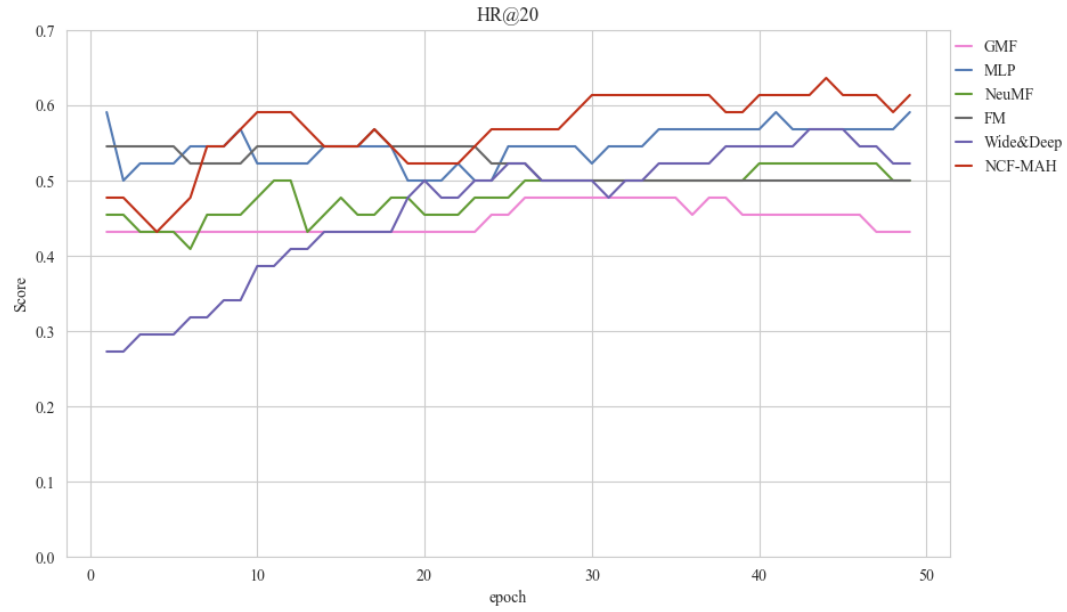

Figure 9. K = 20, Performance of different algorithmic models for HR on the EdX dataset.

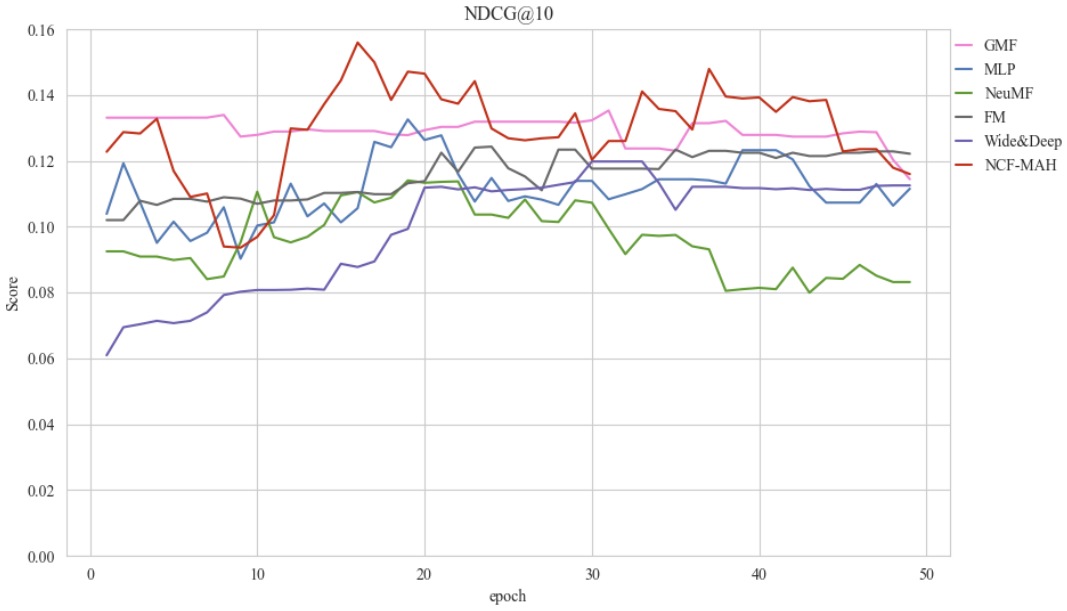

Figure 10. K = 10, Performance of different algorithmic models for NDCG on the EdX dataset.

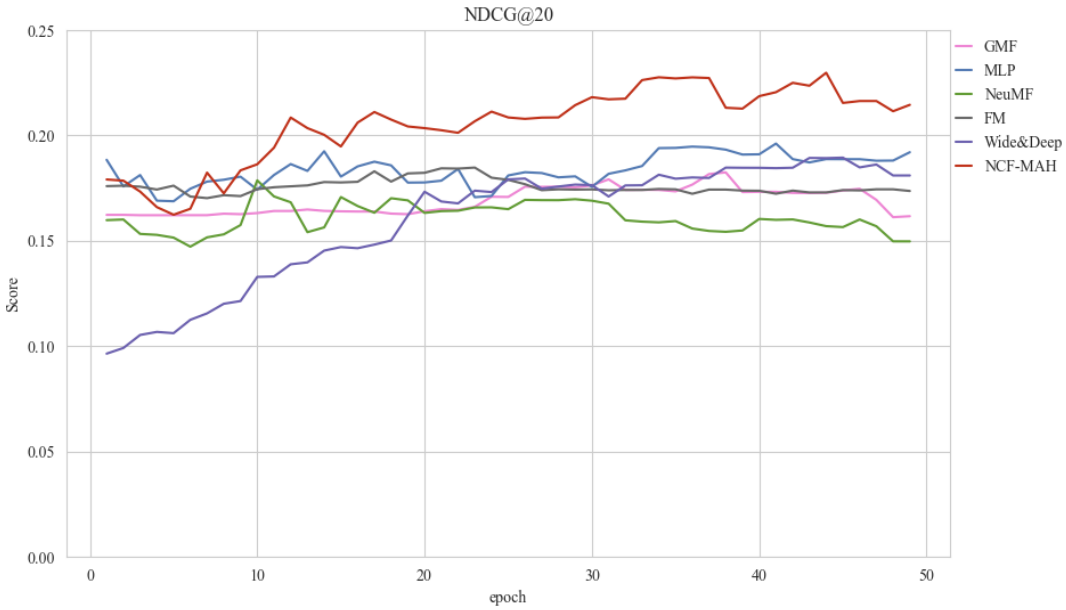

Figure 11. K = 20, Performance of different algorithmic models for NDCG on the EdX dataset.

Table 4. Performance comparison of various models on the MOOC dataset.

Model	MOOC			
	HR@10	HR@20	NDCG@10	NDCG@20
MLP	0.612	0.802	0.332	0.800
GMF	0.105	0.185	0.038	0.188
NeuCF	0.532	0.706	0.302	0.708
FM	0.014	0.173	0.266	0.311
Wide&Deep	0.549	0.790	0.231	0.255
NCF-MAH	**0.662**	**0.804**	**0.405**	**0.810**

From Tables 4 and 5, it is more intuitive to see that under the same experimental parameters for both datasets, the NCF-MHA model exhibits superior performance metrics. This is due to the consideration of users' implicit interaction data, and through the multi-head attention mechanism, combined with the use of BCE as the model's loss function, it better captures user behavior and improves recommendation accuracy and performance.

Table 5. Performance comparison of various models on the EdX dataset.

Model	EdX			
	HR@10	HR@20	NDCG@10	NDCG@20
MLP	0.213	0.435	0.105	0.402
GMF	0.200	0.387	0.167	0.387
NeuCF	0.228	0.575	0.103	0.588
FM	0.295	0.568	0.124	0.185
Wide&Deep	0.273	0.569	0.120	0.189
NCF-MAH	**0.385**	**0.703**	**0.178**	**0.697**

5.2. Ablation experiments

To test the contribution of each component in the NCF-MAH model to the recommended performance, it was evaluated by the following ablation experiments on both datasets.

In order to deeply investigate the effects of different layer architectures of the multilayer perceptron on the performance of the neural collaborative filtering and multi-head attention mechanism fusion model, we carried out detailed experiments on two datasets. We set up three different MLP layer architectures, namely (128, 64, 32), (256, 128, 64, 32), and (64, 32, 16), and the corresponding models are denoted as NCF-MAH_128_64_32, NCF-MAH_256_128_64_32, and NCF-MAH_64_32_16, and the performance of these models is evaluated by the hit rate and normalized discounted cumulative gain in the Top-10 and Top-20 recommended scenarios to evaluate the model performance, and the results are shown in Tables 6 and 7.

Table 6. Performance of models with different MLP layers on the EdX dataset.

Model	EdX			
	HR@10	HR@20	NDCG@10	NDCG@20
NCF-MAH_128_64_32	0.190	0.409	0.032	0.113
NCF-MAH_256_128_64_32	0.227	0.454	0.093	0.150
NCF-MAH_64_32_16	**0.270**	**0.523**	**0.132**	**0.201**

Table 7. Performance of models with different MLP layers on the MOOC dataset.

Model	MOOC			
	HR@10	HR@20	NDCG@10	NDCG@20
NCF-MAH_128_64_32	0.450	0.633	0.255	0.603
NCF-MAH_256_128_64_32	0.425	0.618	0.243	0.687
NCF-MAH_64_32_16	**0.622**	**0.750**	**0.401**	**0.810**

The experimental results clearly show that the differences in the MLP layer architectures have a significant impact on the performance of the NCF-MAH model on both datasets. Among them, the NCF-MAH_64_32_16 model with the (64, 32, 16) architecture shows obvious advantages, which fully highlights the superiority of this architecture.

On the MOOC dataset, the NCF-MAH_64_32_16 model performs well with significant performance improvement. Compared with the NCF-MAH_128_64_32 and NCF-MAH_256_128_64_32 models, the NCF-MAH_128_64_32 model achieves considerable improvement in the recommendation hit rate measured by the HR@10 and HR@20 metrics, as well as the quality of the recommendations embodied by the NDCG@10 and NDCG@20 metrics. This indicates that the architecture can capture the interaction information between users and items more accurately and effectively improve the accuracy and effectiveness of recommendations. Similarly, the NCF-MAH_64_32_16 model also performs well on the EdX dataset. Compared with the NCF-MAH_128_64_32 and NCF-MAH_256_128_64_32 models, there is a significant improvement in the hit rate and recommendation quality indexes, which further proves the adaptability and effectiveness of the (64, 32, 16) architecture on different datasets.

Combining the experimental results of the two datasets, the (64, 32, 16) architecture excels in balancing model complexity and performance, and is able to efficiently capture the complex

interaction information between users and items, thus significantly improving the accuracy and quality of recommendations.

In addition, three variant models of NCF-MAH are described, namely PureMLP, PureMF, and NCF, and these three variant models are compared with the NCF-MAH model. The specific results are shown in Tables 8 and 9.

Table 8. Ablation experiments on the EdX dataset.

Dataset		Precision@10	Precision@20
EdX	PureMLP	0.025	0.026
	PureMF	0.017	0.021
	NCF	0.027	0.028
	NCF-MAH	**0.032**	**0.030**

Table 9. Ablation experiments on the MOOC dataset.

Dataset		Precision@10	Precision@20
MOOC	PureMLP	0.051	0.035
	PureMF	0.046	0.022
	NCF	0.050	0.032
	NCF-MAH	**0.060**	**0.039**

1) Comparing the performance of the PureMLP algorithm and the NCF-MAH algorithm on the Precision@10 and Precision@20 recommendation metrics, the algorithms that use the matrix decomposition and multi-head attention mechanism in personalized recommendation improve 28% and 15% on the EdX dataset compared to the PureMLP algorithm that uses only multi-layer perceptron, and an 18% and 11% improvement on the MOOC dataset, respectively. Thus, the personalized recommendation algorithm proposed in this paper represents the user-resource interaction in a way that allows for a more comprehensive learning of the user's behavior.

2) Comparing the performance of the PureMF algorithm and the NCF-MAH algorithm on Precision@10 and Precision@20 recommendation metrics, the algorithms that use a multilayer perceptron and multi-head attention mechanism in personalized recommendation improve 28% and 15% on the EdX dataset compared to the PureMF algorithm that only uses matrix decomposition, while on the MOOC dataset by 30% and 70%, respectively. Therefore, the personalized recommendation algorithm proposed in this paper can better improve the recommendation accuracy by capturing higher-order features between users and resources.

3) Comparing the performance of the NCF algorithm and the NCF-MAH algorithm on Precision@10 and Precision@20 recommendation metrics, the addition of a multi-attention mechanism to personalized recommendations improves the performance of the NCF algorithm over the traditional NCF recommendation algorithm by 19% and 7% on the EdX dataset, and by 20% and 22% on the MOOC dataset, respectively. Therefore, the personalized recommendation algorithm proposed in this paper introduces the multi-head attention mechanism, which significantly enhances the ability to capture the complex relationship between users and resources, and thus achieves significant improvement in recommendation accuracy.

It can also be seen from Tables 6 and 7 that the NCF-MAH algorithm, which also considers generalized matrix decomposition, a multilayer perceptron, and a multi-head attention mechanism,

has a greater improvement in the Precision@10 and Precision@20 metrics than the three variants of the algorithm. From the principles of the ablation experiments, it is known that the experimental results are better than the model that only considers the explicit interaction between users and resources by considering the addition of attentional mechanisms and the capture of complex relationships with implicit feedback information about user resources. Therefore, the combined use of training can better recommend resources of interest to the user.

6. Conclusions

In this paper, a personalized recommendation algorithm model of NCF-MAH incorporating implicit feedback is proposed, which combines the ideas of matrix decomposition and multilayer perceptrons, and introduces the multi-head attention mechanism to enhance the model's learning ability and expressive ability. The binary cross-entropy loss function is used for training by optimally adjusting the parameters of the connection layer. The model maps the high-dimensional feature vectors of users and items to the low-dimensional embedding space by deeply analyzing the implicit interaction data between users and resources and using matrix decomposition technology. Combined with the multi-attention mechanism, the model can effectively capture complex feature relationships and ensure the effectiveness and stability of training by optimizing the negative sample selection strategy. Ultimately, the model integrates matrix decomposition and multilayer perceptron methods to improve the prediction accuracy, thus achieving a more accurate personalized recommendation effect. The core advantage of the model is that it can learn the implicit cross-features of users, capture the interaction between low-order and high-order features, further extract user behavioral preferences, improve the prediction performance and the generalization ability of the model, and effectively alleviate the data sparsity problem. Comparison experiments on the MOOC dataset and the EdX dataset show that the model outperforms the comparison model in both HR and NDCG evaluation metrics and achieves good recommendation results, proving the effectiveness of the proposed model.

In the future, we can try to further optimize the overall structure of the model, add a convolutional neural network for contextual feature extraction and learning, and utilize multiple types of data information, such as audio, image, etc., to perform multimodal feature fusion, so as to improve the recommendation effect of its model.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The work was supported by the National Natural Science Foundation of China: 72174079, Lianyungang sixth "521" project: LYG06521202351, Lianyungang Science and Technology Program: CG2325. The material in this paper was not presented at any conference.

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. D. Jannach, Evaluating conversational recommender systems: A landscape of research, *Artif. Intell. Rev.*, **56** (2023), 2365–2400. https://doi.org/10.1007/s10462-022-10229-x
2. Y. Shi, M. Larson, A. Hanjalic, Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges, *ACM Comput. Surv.*, **47** (2014), 1–45. https://doi.org/10.1145/2556270
3. X. Yue, C. Tian'e, The design of personalized learning resource recommendation system for ideological and political courses, *Int. J. Reliab. Qual. Saf. Eng.*, **30** (2023). https://doi.org/10.1142/S0218539322500206
4. G. Honglei, An online education course recommendation method based on knowledge graphs and reinforcement learning, *J. Circuits, Syst. Comput.*, **32** (2023). https://doi.org/10.1142/S0218126623500998
5. V. Narjes, M. Mahdieh, S. Hajar, S. M. Fakhrahmad, Application of k-means clustering algorithm to improve effectivenes of the results recommended by journal recommender system, *Scientometrics*, **127** (2022), 3237–3252. https://doi.org/10.1007/s11192-022-04397-4
6. B. Sinha, R. Dhanalakshmi, DNN-MF: Deep neural network matrix factorization approach for filtering information in multi-criteria recommender systems, *Neural Comput. Appl.*, **34** (2022), 10807–10821. https://doi.org/10.1007/s00521-022-07012-y
7. X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T. Chua, Neural collaborative filtering, in *Proceedings of the 26th International Conference on World Wide Web*, (2017), 173–182. https://doi.org/10.1145/3038912.3052569
8. M. Fu, H. Qu, Z. Yi, L. Lu, Y. Liu, A novel deep learning-based collaborative filtering model for recommendation system, *IEEE Trans. Cybern.*, **49** (2019), 1084–1096. http://dx.doi.org/10.1109/TCYB.2018.2795041
9. Y. Pan, F. He, H. Yu, Learning social representations with deep autoencoderfor recommender system, *World Wide Web*, **23** (2020), 2259–2279. https://doi.org/10.1007/s11280-020-00793-z
10. J. Feng, Z. Xia, X. Feng, J. Peng, RBPR: A hybrid model for the new user cold start problem in recommender systems, *Knowledge-Based Syst.*, **214** (2021), 106732. https://doi.org/10.1016/j.knosys.2020.106732
11. Y. Bai, X. Li, Z. Liu, Y. Huang, T. Guo, M. Hou, et al., csKT: Addressing cold-start problem in knowledge tracing via kernel bias and cone attention, *Expert Syst. Appl*., **266** (2025), 125988. https://doi.org/10.1016/j.eswa.2024.125988
12. M. Alfarhood, J. Cheng, DeepHCF: A deep learning based hybrid collaborative filtering approach for recommendation systems, in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, (2018), 89–96. http://dx.doi.org/10.1109/ICMLA.2018.00021
13. I. Saifudin, T. Widiyaningtyas, Systematic literature review on recommender system: Approach, problem, evaluation techniques, datasets, *IEEE Access*, **12** (2024), 19827–19847. https://doi.org/10.1109/ACCESS.2024.3359274

14. P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in *Proceedings of the Fourth ACM Conference on Recommender Systems*, (2010), 39–46. https://doi.org/10.1145/1864708.1864721
15. C. Chen, M. Zhang, Y. Zhang, Y. Liu, S. Ma, Efficient neural matrix factorization without sampling for recommendation, *ACM Trans. Inf. Syst.*, **38** (2020), 1–28. https://doi.org/10.1145/3373807
16. C. Li, L. Wang, S. Cheng, Enhanced transformer encoder and hybrid cascaded upsampler for medical image segmentation, *Expert Syst. Appl.*, **238** (2024), 121965. https://doi.org/10.1016/j.eswa.2023.121965
17. R. Salakhutdinov, G. Hinton, Multimodal learning with deep boltzmann machines, *Adv. Neural Inf. Process. Syst.*, **24** (2012), 1967–2006. https://doi.org/10.1162/NECO_a_00311
18. Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, *Computer*, **42** (2009), 30–37. https://doi.org/10.1109/MC.2009.263
19. Y. Koren, Factor in the neighbors: Scalable and accurate collaborative filtering, *ACM Trans. Knowl. Discovery Data*, **4** (2010), 1–24. https://doi.org/10.1145/1644873.1644874
20. T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (2015), 1412–1421. https://doi.org/10.18653/v1/D15-1166
21. H. Yang, L. Yao, J. Cai, Y. Wang, X. Zhao, A new interest extraction method based on multi-head attention mechanism for CTR prediction, *Knowl. Inf. Syst.*, **65** (2023), 3337–3352. http://dx.doi.org/10.1007/s10115-023-01867-w
22. J. Yu, G. Luo, T. Xiao, Q. Zhong, Y. Wang, W. Feng, et al., MOOCCube: A large-scale data repository for NLP applications in MOOCs, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (2020), 3135–3142. ttps://doi.org/10.18653/v1/2020.acl-main.285
23. J. Zhang, B. Hao, B. Chen, C. Li, H. Chen, J. Sun, Hierarchical reinforcement learning for course recommendation in MOOCs, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **33** (2019), 435–442. https://doi.org/10.1609/aaai.v33i01.3301435
24. X. He, Z. He, J. Song, Z. Liu, Y. Jiang, T. Chua, NAIS: Neural attentive item similarity model for recommendation, *IEEE Trans. Knowl. Data Eng.*, **30** (2018), 2354–2366. http://dx.doi.org/10.1109/TKDE.2018.2831682
25. S. Juneja, A. Nauman, M. Uppal, D. Gupta, R. Alroobaea, B. Muminov, et al., Machine learning-based defect prediction model using multilayer perceptron algorithm for escalating the reliability of the software, *J. Supercomput.*, **80** (2024), 10122–10147. http://dx.doi.org/10.1007/s11227-023-05836-6
26. X. Sun, H. Zhang, M. Wang, M. Yu, M. Yin, B. Zhang, Deep plot-aware generalized matrix factorization for collaborative filtering, *Neural Process. Lett.*, **52** (2020), 1983–1995. https://doi.org/10.1007/s11063-020-10333-5
27. A. Pujahari, D. Sisodia, Item feature refinement using matrix factorization and boosted learning based user profile generation for content-based recommender systems, *Expert Syst. Appl.*, **206** (2022). https://doi.org/10.1016/j.eswa.2022.117849

28. J. Bobadilla, R. Lara-Cabrera, Á. González-Prieto, F. Ortega, DeepFair: Deep learning for improving fairness in recommender systems, *Int. J. Interact. Multimedia Artif. Intell.*, **6** (2021), 86–94. https://doi.org/10.9781/ijimai.2020.11.001

AIMS Press