Computing and Informatics, Vol. 44, 2025, 176–201, doi: 10.31577/cai_2025_1_176

COLLABORATIVE FILTERING ALGORITHM BASED ON DEEP DENOISING AUTO-ENCODER AND ATTENTION MECHANISM

Zixuan Han, Leilei Shi*, Qiang Sun, Xiuliang Huang, Bing Lei

School of Computer Science and Communication Engineering
Jiangsu University, Zhenjiang, China
&
Jiangsu Key Laboratory of Security Technology for Industrial Cyberspace
Jiangsu University, Zhenjiang, China
e-mail: leileishi@ujs.edu.cn

Lu Liu*

Department of Computer Science
University of Exeter, Exeter, UK
e-mail: l.liu3@exeter.ac.uk

Yao Lu

School of Computer Science and Technology
Anhui University, Anhui, China
e-mail: luyao@ahu.edu.cn

Abstract. The burgeoning of e-commerce and online platforms has led to an explosion in data volume and diversity of user preferences, making effective recommendation systems crucial for personalizing user experiences. While collaborative filtering algorithms are traditionally favoured for their ability to leverage user-item interactions, they grapple with data sparsity and noise challenges. To tackle these challenges, Various approaches have emerged in recent years to tackle these chal-

* Corresponding authors

lenges. Recent strides in deep learning, particularly autoencoders and neural networks, have shown promise in addressing these issues. However, limitations persist, such as suboptimal feature extraction and the underutilization of combined nonlinear and linear latent features in traditional autoencoders, as well as the overlooked impact of active users in recommendations. Addressing these research gaps, this study introduces a novel recommendation algorithm that synergizes a deep denoising autoencoder with an attention mechanism, aiming to refine recommendation performance by mitigating data sparsity and enhancing feature extraction. This fusion approach innovatively combines nonlinear and linear latent features and incorporates a neural attention mechanism, significantly improving the precision and personalization of recommendations. Ultimately, the proposed algorithm's effectiveness is assessed and benchmarked against state-of-the-art approaches, demonstrating its potential to revolutionize recommendation systems by offering more accurate and user-tailored suggestions.

Keywords: Deep learning, denoising auto-encoder, collaborative filtering, attention mechanism, recommendation system

1 INTRODUCTION

The advent of cutting-edge technologies such as Big Data, distributed computing, and the Internet of Things (IoT) has catalyzed an unprecedented expansion in network data [1]. However, this abundance of data brings along the challenge of data overload. To assist users in efficiently accessing the information they need from this vast sea of data, recommendation systems have emerged. When users lack clear search keywords, recommendation systems can analyze their interactive behaviors with items, such as rating, liking, commenting, and sharing, to uncover their potential preferences and needs. By proactively recommending potentially interesting items, these systems aim to minimize the amount of time users invest in searching for pertinent content [2, 3, 4]. Over the past decade, researchers have proposed numerous community detection methods from various perspectives, which are also capable of uncovering users' areas of interest and summarizing key information [5, 6, 7].

The crux of recommendation system research lies in the development of recommendation algorithms. An effective recommendation algorithm not only enhances user satisfaction but also generates significant economic benefits for merchants and enterprises [8]. Consequently, improving user satisfaction and recommendation accuracy assumes crucial importance, with particular focus on addressing data sparsity and cold-start issues within the field of recommendation research. In the early 1990s, Goldberg et al. introduced the collaborative filtering algorithm (CF) [9]. Praised for its interpretability and real-time personalized recommendations, CF attracted considerable attention from experts and scholars worldwide [10].

In recent years, the field of recommendation algorithms has witnessed significant advancements in accuracy, primarily driven by research focusing on relevance within communities and clusters [11, 12, 13]. Furthermore, collaborative filtering algorithms based on matrix decomposition [14] have gained substantial recognition as distinguished recommendation algorithms. However, recommendation algorithms that predominantly rely on the association between users and items often encounter limitations, particularly when user-item interaction data is incomplete. This inadequacy in capturing the full spectrum of users' preferences can lead to a diminution in the accuracy of recommendations. On the one hand, these algorithms solely employ linear models to capture user-item interactions, thus struggling to capture deeper implicit features of users and items [15]. On the other hand, factors such as high-dimensional features and the massive scale of the dataset exacerbate the common issue of sparsity in the user-item rating matrix [16]. Furthermore, these algorithms rely on similarity calculations between users or items for recommendations, making them less effective when dealing with new users or items, ultimately affecting system performance.

Deep learning techniques provide a promising solution to these challenges. Firstly, deep learning models can unveil implicit features within data using neural network architectures, enabling a more comprehensive representation and understanding of the underlying data. Secondly, deep learning allows for the mapping of data with varying dimensions into a shared feature space, facilitating joint feature representations of the data [17]. By incorporating deep learning techniques into traditional recommendation algorithms, it becomes feasible to overcome the challenges faced by conventional methods in capturing intricate user and item features, as well as addressing issues related to data sparsity and cold-start problems [18].

This paper builds upon our previous research [19]. It enhances the dynamic collaborative filtering recommendation algorithm by integrating a deep denoising auto-encoder and introducing an attention mechanism to mitigate the impact of active users on experimental results. This enables the exploration and learning of both linear and nonlinear user and item features. The main contributions of this paper are outlined as follows:

1. Development of an Enhanced Deep Denoising Auto-Encoder Network: This study introduces an enhanced structure of the conventional auto-encoder network to make it more adaptable for representing and learning complex feature patterns. By introducing noise following a Gaussian distribution and then denoising it, we enhance the generalization capability and robustness of the original auto-encoder network. This enhancement effectively tackles problems like overfitting and improves the model's prediction capability and recommendation accuracy.

2. Advanced Feature Analysis Using Enhanced Deep Denoising Auto-Encoder: This study introduces a Deep Denoising Autoencoder Convolution and Multi-Layer Perceptron (DAE-CMLP) algorithm, which extracts and analyzes both linear and nonlinear latent features from the user-item rating matrix. By distilling these features into a lower-dimensional space, we enhance the collaborative filtering algorithm's ability to discern nuanced user-item relationships. This ap-

proach not only addresses the cold-start problem by analyzing user interactions but also integrates an attention mechanism to balance the impact of active users, resulting in more accurate and unbiased recommendations.

3. We conducted experiments on two publicly available datasets to validate the performance of our algorithm. The experimental results demonstrate that our proposed algorithm outperforms the current state-of-the-art comparison algorithms, significantly improving the performance of traditional recommendation algorithms and effectively addressing issues related to data sparsity and insufficient feature extraction.

The remainder of the paper is organized as follows. Section 2 provides a summary and analysis of the current state of research on collaborative filtering algorithms, deep learning models, and hybrid algorithms. Section 3 introduces the improved deep denoising auto-encoder network structure. Section 4 introduces the fusion method of linear and non-linear features of the scoring matrix. Section 5 presents the recommendation algorithm combining deep denoising autoencoder and attention mechanism. Section 6 presents the design of the experiments and the analysis of the results. Section 7 summarises the conclusions and future work.

2 RELATED WORK

By analyzing users' past behaviors and preferences, identifying their latent interests, and providing recommendations aligned with these interests, collaborative filtering rules have found widespread application in research on recommendation systems. They have become a fundamental paradigm in this field and have achieved significant success. In the research conducted by Wu et al. [20], they extensively discuss the developmental journey of recommendation algorithms based on collaborative filtering rules. They also delve into the potential challenges associated with such algorithms and provide corresponding solutions. They posit that the success of this approach is partly attributed to its adeptness at harnessing valuable information from user-item interactions to generate personalized recommendations. Nevertheless, challenges like data sparsity, cold-start issues, and recommendations for new users and items continue to pose difficulties for this method to address the challenges as mentioned above related to cold start and quantifiability, Han et al. [21] proposed a collaborative filtering algorithm that combines user information features and temporal factors to enhance recommendation accuracy. Additionally, Cheng et al. [22] introduced a collaborative filtering hybrid imputation algorithm to tackle information sparsity issues. This algorithm achieves information extraction by filling sparse matrices from both user and item perspectives.

To enhance rating prediction within the recommendation system, the research report [23] introduced the Matrix Factorization (MF) model as the foundational model for boosting the Adaboost algorithm. By setting a threshold, the rating prediction problem can be transformed into a classification problem. This approach involves partitioning the continuous range of ratings into distinct categories, making it easier for the recommendation system to interpret and manage the recommendation results. The experimental results demonstrated enhanced predictive accuracy when compared to the classic algorithm. Additionally, the study [24] integrated auxiliary information into the Bayesian Matrix Factorization (BMF) method, thereby effectively improving the accuracy of recommendation outcomes.

Deep learning has demonstrated substantial potential in artificial intelligence research, with numerous scholars exploring its feature learning capabilities within the realm of recommendation systems [25, 26, 27, 28]. In addition to collaborative filtering, the ascent of deep learning technology has unlocked fresh opportunities in the field of recommendation systems. Deep learning models possess the ability to autonomously acquire intricate feature representations, consequently enhancing their capacity to capture the inherent relationships between users and items. The fusion of deep learning with collaborative filtering has spawned a range of robust hybrid models, further propelling the evolution of recommendation systems. For instance, Sedhain et al. [29] pioneered the application of auto-encoders in collaborative filtering with their AutoRec model, which reconstructs the user-item rating matrix through encoding and decoding processes, minimizing reconstruction errors to train the model. Strub et al. [30] utilized a stacked noise reduction auto-encoder to learn latent representations of users and items. They addressed data sparsity issues by assigning unobserved data in the rating matrix as zero and enhanced model robustness by introducing noise to the rating vector. Wu et al. [31] introduced a collaborative noise reduction auto-encoder model that takes into account user personality differences, introducing personalization factors to augment the performance of personalized recommendations.

In practical scenarios, the user-item interaction function can often be too intricate to be adequately learned through traditional dot products alone. Neural Collaborative Filtering (NCF) [32] employs a multilayer perceptron to grasp the user-item interaction function, thereby enhancing the model's capacity to capture nonlinear relationships. To address the limitations of multilayer perceptrons in capturing linear relationships, NeuMF combines generalized matrix decomposition and multilayer perceptrons into a unified model. Outer Product-based Neural Collaborative Filtering (ONCF) [33] introduces the use of outer products to explicitly model pairwise associations between embedded spatial dimensions, resulting in a more expressive and semantically meaningful two-dimensional interaction map. Building upon this two-dimensional interaction graph, ConvNCF utilizes a convolutional neural network to learn higher-order correlations between embedding dimensions.

3 IMPROVED DEEP DENOISING AUTO-ENCODER NETWORK

In traditional collaborative filtering algorithms, the implicit feature vectors of users and items are typically initialized randomly [34]. However, it is worth noting that there exist numerous nonlinear features within real items that cannot be effectively learned by conventional algorithmic models. Autoencoders can not only reduce and compress high-dimensional data but also learn deep nonlinear features [35]. By stacking multiple encoder and decoder layers, autoencoders can build deep neural networks, thereby more effectively capturing abstract features within the data. Consequently, autoencoders find extensive application and play a significant role in recommendation system research. Hence, we contemplate the utilization of the auto-encoder network to extract nonlinear item features and integrate them into the dynamic collaborative filtering algorithm, with the aim of enhancing the performance of the conventional recommendation model.

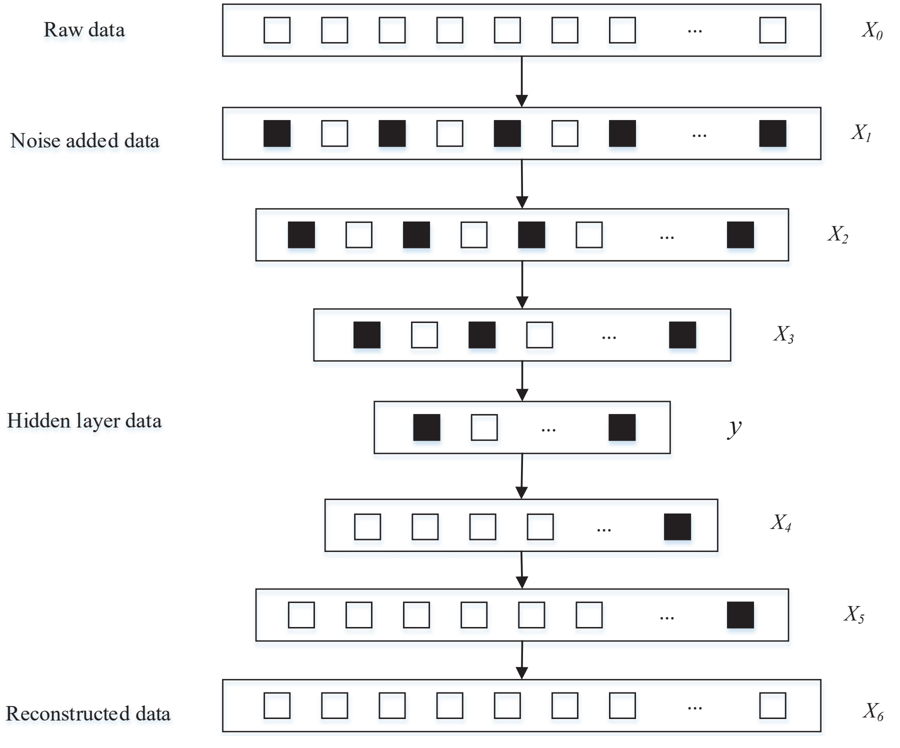


Figure 1. Improved six-layer depth denoising auto-encoder network structure

In order to further enhance the generalization capability and robustness of the traditional auto-encoder model, this section expands upon the original network structure of the conventional three-layer auto-encoder, as illustrated in Figure 1, into a six-layer deep network. The improved six-layer deep denoising autoencoder introduces more hidden layers, increases the network depth, and thereby enhances its capability for representation learning and noise reduction. Each hidden layer is connected to the preceding and succeeding layers, forming a deep encoder-decoder structure. The input data first passes through the encoder section, undergoing layer-by-layer feature extraction and representation learning, and then goes through the decoder section for reconstruction and recovery. The dimensions of the hidden layers gradually decrease and then increase, aiding the network in extracting higher-level abstract features and restoring input data during the decoding process.

In denoising autoencoders, input data is subjected to some noise, such as Gaussian noise or random dropout. By training the network to recover the original noise-free input data, the network can learn useful features within the data and filter out the noisy components. Through the increased depth of the network, the improved six-layer deep denoising autoencoder can better learn the representation of complex data, thereby enhancing denoising performance. This modification allows the backpropagation algorithm to better accommodate the representation and learning of intricate feature patterns, effectively mitigating issues such as overfitting.

Encoding stage: X_0 is the original user item scoring matrix, and Gaussian noise is added to X_0 with the noise rate $\tau \in [0,1]$, and the black data in the figure is the added Gaussian noise. After adding Gaussian noise to ensure that the expected value of the original scoring matrix does not change, the original user item scoring matrix needs to be zeroed, specifically the probability of containing noisy data in the scoring matrix is set to zero, that is, the data other than noise is magnified $\frac{1}{1-\tau}$ times, X_1 is the final user item scoring matrix after the noise addition process, as shown in Equation (1).

$$X_1 = \left\{ X_1 \mid X_1 = \frac{X_0}{1-\tau} \right\}. \tag{1}$$

The noise-added high-dimensional user-item rating matrix X_1 is used as input, and the low-dimensional dense matrix of the user's implied feature vector y is computed through the hidden layer with a dimensionality reduction and compression operation, as shown in Equation (2):

$$y = f\left(W \cdot X_S + b\right), \tag{2}$$

where y is the low-dimensional dense matrix of the hidden layer; $f(\cdot)$ is the sigmoid activation function $f(x) = \frac{1}{1+e^{-x}}$; W is the $m \times n$ dimensional weight matrix; s is the number of layers corresponding to the model and b is the offset at encoding.

Decoding stage: the noise-added high-dimensional scoring data is compressed and downscaled by the hidden layer y to obtain the low-dimensional scoring data X_4, and then reconstructed by the three-layer decoder to produce the output matrix X_6 with the same scale as the original scoring matrix, as shown in Equation (3):

$$X_S = f'\left(W' \cdot y + b'\right), \tag{3}$$

where X_S is the data at layer s of the decoding stage; $f'(\cdot)$ is the mapping function; W' is the weight matrix of $n \times m$ dimensions; b' is the offset at decoding.

The objective function of the improved depth denoising auto-encoder is shown in Equations (4) and (5) based on the original scoring matrix X_0 and the reconstruction matrix X_6 to reconstruct the error between the two.

$$\arg\min_{W,W',b,b'} L\left(X_0, X_6\right) + R\left(W, W', b, b'\right), \tag{4}$$

$$L\left(X_0, X_6\right) = \frac{1}{N}\sum_{i=1}^{N}\left\|X_0 - X_6\right\|_2^2, \tag{5}$$

where L is the loss function of the depth-denoising auto-encoder; $R\left(W, W', b, b'\right)$ is the regular term, which controls the complexity of the model; N is the number of elements of the user-item scoring matrix.

The improved depth-denoising auto-encoder network augments the depth of the conventional model and introduces noise to further enhance the generalization capability and robustness of the original conventional auto-encoder network.

4 LINEAR AND NONLINEAR FEATURE FUSION OF SCORING MATRICES

In traditional recommendation algorithms, implicit feature vectors of users and items are typically initialized randomly. However, real-world items often possess not only observable linear features but also unobservable nonlinear implicit features. The deep denoising autoencoder model is capable of learning the nonlinear features of rating data, while the dynamic collaborative filtering algorithm excels at extracting the linear features of users and items. Consequently, in this section, we explore how to integrate these two types of features to address the challenge faced by traditional recommendation algorithms in capturing the deeper latent features of users and items.

To begin with, an enhanced deep denoising autoencoder network is employed to acquire the nonlinear features from the user-item scoring matrix. Figure 2 illustrates the model that combines linear and nonlinear features within the scoring matrix. By inputting the original user-item scoring matrix into the enhanced deep denoising autoencoder network, we can achieve dimensionality reduction of high-dimensional sparse scoring data, as illustrated in Equation (6):

$$q'' = f\left(W \cdot X + b_n\right), \tag{6}$$

where q'' is the low-dimensional nonlinear feature vector of the user and item after compression by the depth-denoising auto-encoder; f is the activation function of the depth-denoising auto-encoder; W represents the weight matrix for both encoding and decoding processes, while X denotes the initial feature vector of the item. Additionally, b_n signifies the offset.

Subsequently, the low-dimensional nonlinear features resulting from the depth denoising autoencoder compression process are merged with the item-based collaborative filtering algorithm, as illustrated in Equation (7):

$$\hat{r}_{ui} = \mu + b_u + b_i + \gamma \cdot q'' + q', \tag{7}$$

where $\hat{r}_{ui}$ is the predicted score of user u for item i; μ is the average of the scores; b_u and b_i are the offset values relative to user u and item i; γ is the hyperparameter,

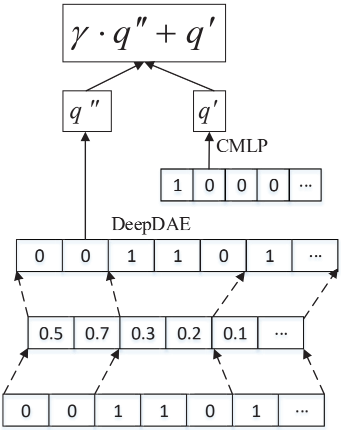


Figure 2. Linear and nonlinear feature fusion model for scoring matrix

which mainly controls the weights of the feature vectors extracted by the depth-denoising auto-encoder; q' is the linear feature vector of user and item learned by the collaborative filtering model.

A dynamic collaborative filtering algorithm is employed to extract the linear features from the user-item rating matrix. The chosen optimization objective is the CMLP model, which is a deep dynamic collaborative filtering algorithm [19]. This model enhances MLP-based collaborative filtering by incorporating explicit and implicit feedback as well as local associations. It achieves this by incorporating embedded implicit similarity groups to calculate user and item similarities. The model is represented in Equation (8):

$$q' = [a_n,\ b_n,\ c_n]\,, \tag{8}$$

where a_n is the connection processing of two vectors of users and items, b_n is the higher-order interaction vector obtained from implicitly similar user groups and items after MLP learning and c_n is the higher-order interaction vector between users and implicitly similar item groups; $[a_n,\ b_n,\ c_n]$ means connecting the three vectors.

The improved deep denoising auto-encoder effectively captures the nonlinear features of both users and items, as exemplified in Equation (9). The CMLP model integrates the prediction function of the enhanced deep denoising auto-encoder. Consequently, the collaborative filtering model, incorporating the deep denoising auto-encoder, can extract nonlinear item features using the enhanced deep denoising auto-encoder, alongside linear features using the dynamic collaborative filtering algorithm. Furthermore, the user's implicit features are represented by the items they have historically interacted with, which mitigates the cold start problem to some extent.

$$\hat{r}_{ui} = \mu + b_u + b_i + \gamma \cdot q'' + [a_n,\ b_n,\ c_n]\,. \tag{9}$$

5 A RECOMMENDATION ALGORITHM INCORPORATING DEEP DENOISING AUTO-ENCODER AND ATTENTION MECHANISM

While the collaborative filtering model incorporating the deep denoising auto-encoder effectively captures both linear and nonlinear user features, it falls short in distinguishing between various historical interaction items and fails to account for a user's preference for different attribute items. For instance, when predicting a user's preference for a disaster movie, the historical interactions with disaster movies should carry more weight, whereas when predicting the user's preference for a costume movie, the weight of disaster movies in historical interactions should decrease. Additionally, ignoring the influence of active users on recommendation results makes it challenging to recommend diverse topics to users, ultimately affecting the overall recommendation quality.

In order to address this issue, it is necessary to emphasize the varying impact of a user's historical interactions with different items on the prediction results and distinguish the degree of influence of different historical interaction items. Inspired by the work of He et al. [36], the introduction of a neural network attention mechanism can effectively address this limitation in the model. Therefore, the collaborative filtering model incorporating the depth denoising auto-encoder is integrated into the neural network attention weight W_{ij}. This results in the final recommendation algorithm that combines the depth-denoising auto-encoder and the attention mechanism. The prediction function of this model is presented in Equation (10):

$$\hat{r}_{ui} = \mu + b_u + b_i + \gamma \cdot q'' + W_{ij}\,[a_n,\ b_n,\ c_n]\,. \tag{10}$$

Here, W_{ij} represents the weight assigned to item j which user u has previously interacted with, in the context of predicting the user u's preference for the target item i. However, there is a special case here when there is no user in the training set that has interacted with item i and item j, then the weight W_{ij} cannot be obtained. To solve this problem, we can consider linking q_j with the fused features $\gamma \cdot q_i'' + q_i'$ obtained by the improved deep denoising auto-encoder and the collaborative filtering algorithm, where q_j is the implied eigenvector of the interacted item j vector and redefining the weights by the fused features of the target item i with the features of the already interacted item j.

Figure 3 illustrates the network architecture of the attention model. In this paper, the primary function of this attention mechanism is to compute the similarity between the target item i and the historical interaction item j. It accomplishes this by performing the dot product of the feature vectors of the target item i and historical interaction item j, which is then input into a three-layer MLP network for training, ultimately yielding the attention weight W_{ij}. The experiment employs the scaled exponential linear unit (SeLU) as the model's activation function, as depicted in Equation (11). This function is an enhanced variant of the exponential linear unit (eLU) and includes its own normalization mechanism [36]. The attention weight and the attention function are presented in Equation (12).

$$SeLu(x) = \alpha \begin{cases} \varepsilon(e^x - 1), & x < 0, \\ x, & x > 0, \end{cases} \tag{11}$$

$$\begin{cases} W_{ij} = \dfrac{\exp\left[f\left(\left(q_i' + \gamma \cdot q_i''\right), q_j\right)\right]}{\left[\sum \exp\left[f\left(\left(q_i' + \gamma \cdot q_i''\right), q_j\right)\right]\right]^{\xi}}, \\ f\left(\left(q_i' + \gamma \cdot q_i''\right), q_j\right) = SeLU\left[w\left(\left(q_i' + \gamma \cdot q_i''\right) \odot q_j\right) + b\right]. \end{cases} \tag{12}$$

From the literature [1], $\alpha \approx 1.0507$ and $\varepsilon \approx 1.6733$ are the optimal parameter configurations for this activation function. ξ is the smoothing exponent, which is used to reduce the over-punishment of active users. When $\xi = 1$, the formula is a softmax function that normalizes the weights, and when $0 < \xi < 1$, active users are not over-punished. w is the matrix of weights for the input projection to the hidden layer and b is the offset.

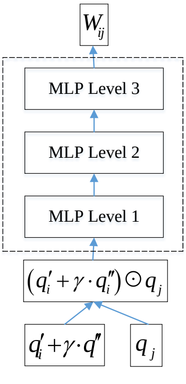


Figure 3. Network structure of the attentional model

In the implicit feedback model, learning can be viewed as a binary problem. This involves defining the user-item interaction history set as positive instances and non-interactions as negative instances. Predicted scores are normalized using the Sigmoid function to obtain the probabilistic output. Subsequently, the loss is calculated through cross-entropy, combining the deep denoising auto-encoder and the attention mechanism. The loss function for the recommendation algorithm is presented in Equation (13):

$$L = -\frac{1}{N}\left[\sum \log \delta\left(\hat{r}_{ui}\right) + \sum \log\left(1 - \delta\left(\hat{r}_{ui}\right)\right)\right] + \lambda \left\|\theta\right\|^2, \tag{13}$$

where N is the number of training sets; δ is the probability of possible interaction between user u and target item i, obtained from the sigmoid function; λ is the regular term coefficient.

Figure 4 illustrates the recommendation algorithm model that combines the deep denoising autoencoder and the attention mechanism. The logical relationships between parameters are depicted as shown in the upper-left corner, with thick solid lines representing vector inner product relationships, and thin dashed lines and thick dashed lines denoting the network's inputs and outputs, respectively.

The model takes the prediction of the user u's preference for target item number 1 as an example, the lower left side of the figure indicates a row vector of the user u's historical interaction matrix, q_2 indicates the implicit feature vector of the user u's interaction item number 2, S_{12} indicates the similarity between target item 1 and interaction item 2, W_{12} is the attention weight of target item 1 and interaction item 2 obtained by neural network training, and other parameters are the same. q_1'' denotes the nonlinear feature vector of target item 1 learned by the deep denoising auto-encoder. q_1' denotes the linear feature vector of target item 1 mined by the CMLP model.

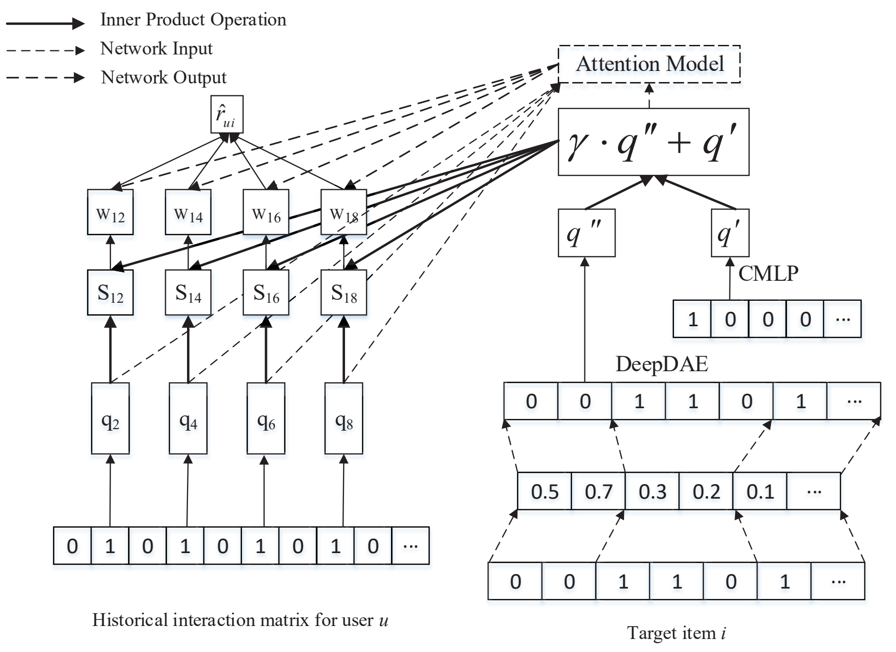


Figure 4. A recommendation algorithm incorporating deep denoising auto-encoder and attention mechanism

6 EXPERIMENTS

6.1 Experimental Environment

The experiment was conducted on a personal computer equipped with an NVIDIA GeForce GTX 1650 graphics card, 16 GB of RAM, and an AMD Ryzen 5 3500X processor. TensorFlow, a deep learning framework developed by the Google AI team, was utilized for development, with programming carried out in Python. Detailed parameter configurations can be found in Table 1.

Name	Configuration
CPU	AMD Ryzen 5 3500X 6-Core Processor 3.59 GHz
GPU	NVIDIA GeForce GTX 1650
RAM	16 GB
Operating System	Windows 10 Professional
Programming Languages	Python 3.9
Deep Learning Framework	TensorFlow 2.11.0

Table 1. Experimental environment parameters

6.2 Data Set

In this experiment, two widely employed public datasets within the field of recommendation systems were chosen: the MovieLens dataset [37] and Pinterest [38]. The MovieLens dataset is sourced from a platform dedicated to aggregating user movie reviews and offering movie recommendations, while Pinterest is an image-centric platform where users can share images of their interests. Specifically, the MovieLens-1M dataset comprises a substantial one million movie reviews, each rated on a scale of 1 to 5, resulting in a dataset density of 4.19 %. To accurately reflect user-movie interactions, preprocessing steps were applied to the MovieLens dataset, as outlined in Equation (14). Precisely, this operation entails the transformation of explicit ratings within the user-movie interaction matrix into implicit ratings. Implicit ratings are designated as 1 when a user has interacted with a movie and 0 when there is no interaction. r_{ui} indicates the user rating of the item.

$$\begin{cases} 1, & r_{ui} > 0, \\ 0, & r_{ui} = 0. \end{cases} \tag{14}$$

The Pinterest dataset is centered around image recommendations on a social networking site. This dataset is extensive yet sparse, with over 20 % of users having only performed a single "pin" action. Consequently, preprocessing of the Pinterest dataset involves retaining users with 20 or more pin operations. The statistical information regarding the experimental dataset is presented in Table 2.

Data Set	Users	Items	Ratings	Sparsity
MovieLens 1M	6 040	3 706	1 000 000	95.80 %
Pinterest	55 187	9 916	1 500 809	99.70 %

Table 2. Data set information

The experiments are categorized according to the timestamps provided in the dataset. The test set comprises the most recent historical rating records for each user, excluding this data from the training set. This process involves combining each test set with 99 other randomly sampled training sets. Each algorithm generates predicted scores for 100 movies, and the evaluation of each test set is performed using HR and NDCG metrics.

6.3 Comparative Algorithms and Evaluation Indicators

In order to evaluate the performance of this algorithm, several traditional algorithms and more advanced algorithms of recent years were selected for comparison experiments, and the experimental comparison model is as follows:

1. FISM [39]: Recommendation algorithm based on item similarity. This algorithm leverages a user's historical interaction data to indirectly incorporate singular value decomposition techniques into an item-based similarity algorithm.
2. Knn-basic [40]: Standard neighborhood-based collaborative filtering based on a user-based approach.
3. NeuMF (Neural Matrix Factorization) [41]: Combining the two models of generalized matrix factorization and MLP, which is the model with the strongest performance in the promotion of the NCF framework.
4. ConvMF (Convolutional Matrix Factorization) [42]: Fusion of convolutional neural networks and PMF.
5. SVD++ [43]: An optimized Singular Value Decomposition (SVD) algorithm has been developed to enhance prediction accuracy through the generation of implicit feedback.
6. CMLP [19]: A collaborative filtering model has been created, combining Convolutional Neural Networks with a Multi-layer Perceptron. This integrated approach effectively captures both local correlations and explicit as well as implicit feedback information.
7. DAE-CMLP: A recommendation algorithm that integrates a deep denoising auto-encoder and an attention mechanism.

To assess the performance of the recommendation model, standard evaluation metrics in the recommendation algorithm, namely NDCG (Normalized Discounted Cumulative Gain) and HR (Hit Rate), were employed as indicated in Equation (15). Here, HR represents the proportion of items recommended in the final list of recommendations compared to the total number of items in the entire test set. This metric provides insight into the percentage of recommended items within the overall test set.

$$HR@n = \frac{NumberofHits@n}{TestSets}. \tag{15}$$

As shown in Equation (16), the ranking of the recommendation list is represented by the cumulative gain CG_n. Since the actual recommendation needs to rank the results with high relevance to the recommendation first, it is necessary to introduce a position influence factor on top of CG_n, which is represented by the discounted cumulative gain DCG_n as shown in Equation (17). The best-recommended result returned by the user in the recommendation list is the maximum value of DCG_n, denoted by $IDCG_n$. This is normalized so that the final Normalised Discounted Cumulative Gain (NDCG) is shown in Equation (18).

$$CG_n = \sum_{i}^{n} rel_i, \tag{16}$$

$$DCG_n = \sum_{i=1}^{n} \frac{2^{rel_i} - 1}{\log_2{(i+1)}}, \tag{17}$$

$$NDCG_n = \frac{DCG_n}{IDCG_n}. \tag{18}$$

6.4 Experimental Protocol and Parameter Settings

The experiments were conducted using a recommendation model that takes into account both explicit and implicit feedback. The algorithm can be outlined in the following steps.

Step 1: The dataset has undergone preprocessing. To investigate the interaction between users and items, it was imperative to transform the data from explicit ratings to implicit ratings. In this transformation, instances, where users interacted with items in the dataset, were assigned a value of 1, while instances without interaction were assigned a value of 0.

Step 2: The item information from the user-item dataset is fed into a modified deep denoising auto-encoder for training. The backpropagation algorithm is employed for multiple iterations to learn the nonlinear implicit feature vector of the items.

Step 3: Input the obtained non-linear feature vectors of the items to the FISM model for feature fusion, and then input the fused features to the attention network, and adjust their weights by the attention network.

Step 4: The prediction value of the target item is computed based on the attention weight. Subsequently, the output prediction score is normalized using the Sigmoid function to derive the probability output of the prediction score. Following this, the loss is calculated using cross-entropy.

To evaluate the overall performance of the proposed new algorithm, we conducted experiments in three different scenarios, which are detailed as follows:

Scenario 1: Evaluate the performance of various algorithms on the same dataset by comparing performance metrics, including NDCG@5, NDCG@10, HR@5, and HR@10, across different algorithms on both the MovieLens and Pinterest datasets.

Scenario 2: Evaluate the performance of a specific algorithm on diverse datasets by comparing its performance metrics, including NDCG@5, NDCG@10, HR@5, and HR@10, across two distinct datasets, MovieLens and Pinterest.

Scenario 3: Study the impact of this paper's attention mechanism on recommendation performance. A baseline algorithm is selected and iterated 50 times on the MovieLens dataset and the Pinterest dataset respectively to compare the NDCG@10 and HR@10 performance evaluation metrics of this paper with and without attention.

All three schemes of the experiment were compared at the same parameter settings by default, as shown in Table 3 for the default parameter settings of the improved depth denoising auto-encoder network.

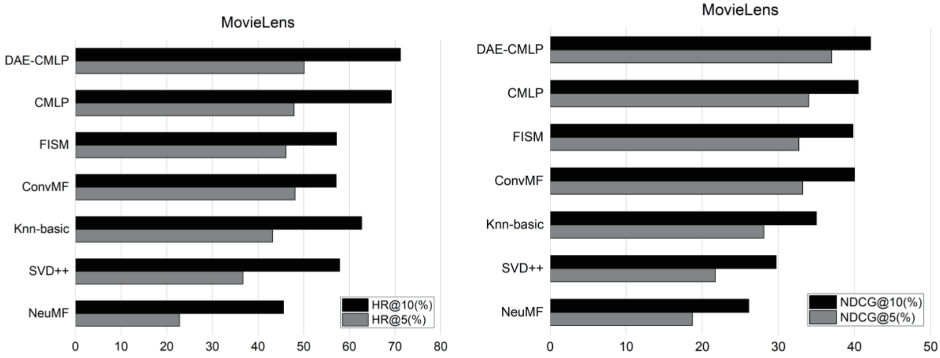
Parameter Name	Parameter Values
Number of neurons	1 024, 512, 256
Noise addition rate (ρ)	5
Learning rate 1 (lr_1)	0.1
Embedding_size	16
Smoothing index (ξ)	0.45
Number of negative feedback (num_neg)	4
Learning rate 2 (lr_2)	0.01
Number of iterations (epochs)	50

Table 3. Experimental default parameter settings

6.5 Analysis of Experimental Results

To comprehensively demonstrate the performance and applicability of our algorithm in this study, we conducted experiments from two aspects: comparing different algorithms on the same dataset and contrasting the same algorithm on different datasets. We computed the NDCG and HR evaluation metrics under both 5 and 10 recommendation conditions. The bar charts for NDCG and HR evaluation metrics are depicted in Figures 5 and 6, respectively.

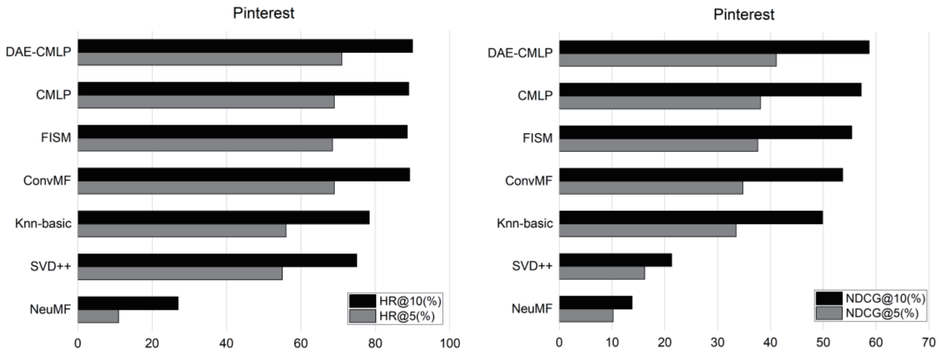
First, the performance of different algorithms on the same dataset is compared. Here we take the classic SVD++ algorithm and the more recent FISM algorithm as examples. With 5 and 10 recommendations, the HR of DAE-CMLP on the MovieLens dataset is 14.16 % and 13.45 % higher than that of the SVD++ model, and 4.2 % and 4.25 % higher than that of the FISM model, respectively. With 5 and 10 recommendations, the HR of DAE-CMLP on the Pinterest dataset is 19.26 % higher than that of the SVD++ algorithm. The HR of DAE-CMLP is 19.26 % and 20.32 % better than the SVD++ algorithm, and 2.27 % and 1.95 % better than the FISM model, respectively.



a) HR evaluation metrics for the MovieLens dataset

b) NDCG evaluation metrics for the MovieLens dataset

Figure 5. HR@10 and NDCG@10 metrics for the first 50 iterations of the MovieLens dataset



a) HR evaluation metrics for the Pinterest dataset

b) NDCG evaluation metrics for the Pinterest dataset

Figure 6. HR@10 and NDCG@10 metrics for the first 50 iterations of the Pinterest dataset

With 5 and 10 recommendations, the NDCG of DAE-CMLP on the MovieLens dataset was 15.23 % and 12.67 % better than the SVD++ algorithm, and 4.28 % and 2.76 % better than the FISM algorithm, respectively. The NDCG of DAE-CMLP on the MovieLens dataset was 25.55 % and 37.74 % better than the SVD++ algorithm, and 4.23 % and 3.29 % better than the FISM algorithm, respectively.

Then, the performance of the same algorithm on different datasets is compared. Taking DAE-CMLP as an example, the performance of DAE-CMLP on the MovieLens dataset is generally lower than that on the Pinterest dataset, both in terms of HR and NDCG evaluation metrics. The reason for this situation may be related to the sparsity of the dataset and the number of samples. During data pre-processing, the difference in sample density between the two datasets was caused by the fact that the Pinterest dataset was highly sparse, retaining only users with pin operations greater than or equal to 20.

DAE-CMLP significantly outperforms the classical SVD++ algorithm and the famous FISM algorithm in terms of NDCG@5, NDCG@10, and HR@5, HR@10 recommendation performance indicators on both MovieLens and Pinterest datasets, and outperforms the original improved algorithm CMLP, which can fully illustrate the performance of the attention mechanism and the advancement of the algorithm in this paper.

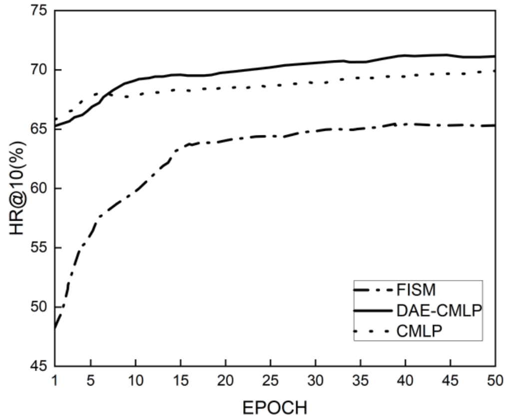
To illustrate the performance of the attention mechanism in this paper, FISM was chosen as the baseline algorithm and the HR and NDCG evaluation metrics were compared between DAE-CMLP and CMLP for the first 50 iterations of training on two different datasets. The default number of recommendations for this experiment is Top_10, i.e. the top 10 items are recommended to the target user.

As shown in Figures 7 and 8, the HR and NDCG evaluation metrics for the first 50 iterations of the three compared algorithms on two different datasets, MovieLens and Pinterest, can be seen from the curves that DAE-CMLP significantly outperforms both the traditional FISM model and CMLP in both HR and NDCG evaluation metrics, indicating that the performance of CMLP can be effectively improved by introducing deep denoising auto-encoders to extract the non-linear features of the data.

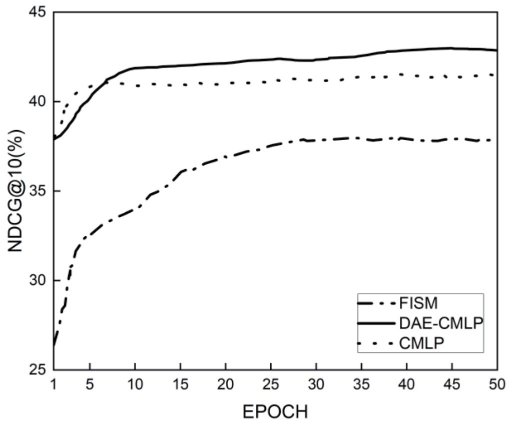
The CMLP algorithm outperforms the DAE-CMLP algorithm for about the first 8 iterations of the MovieLens dataset and about the first 6 iterations of the Pinterest dataset, which is since the attention model requires a process of parameter training. With the MovieLens dataset and the default optimal parameter settings, DAE-CMLP outperformed CMLP by 1.51 % and 2.12 % in the HR@10 and NDCG@10 evaluation metrics, respectively. The two algorithms converge quickly and iteratively in the first 10 training iterations, and then slowly increase until they level off. As the training iterations progress, the performance of DAE-CMLP consistently improves and stabilizes across all metrics. This observation strongly supports the notion that the incorporation of the attention mechanism effectively enhances to the recommendation performance.

7 CONCLUSION AND FUTURE WORK

This study enhances traditional three-layer autoencoders by introducing a six-layer deep denoising autoencoder, significantly improving the network's generalization ability and robustness. This advanced architecture facilitates more effective learning



a) HR@10 evaluation metrics for the MovieLens dataset

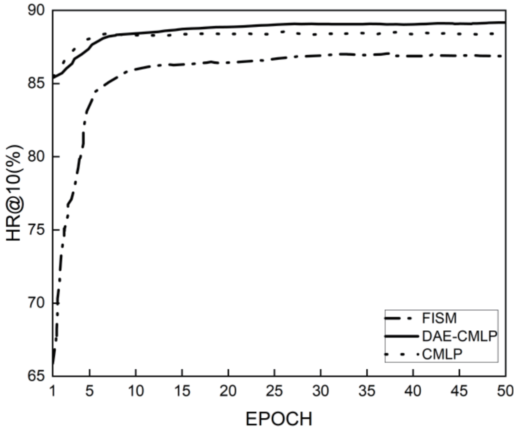


b) NDCG@10 evaluation metrics for the MovieLens dataset

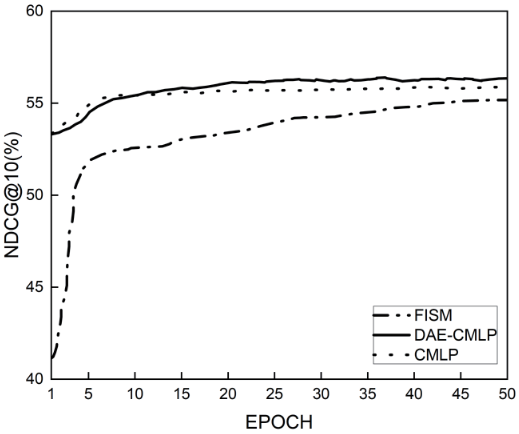
Figure 7. HR@10 and NDCG@10 metrics for the first 50 iterations of the MovieLens dataset

of latent item features from the user rating matrix, addressing challenges related to data sparsity and sparse rating matrices. Consequently, it refines the performance of the recommendation system.

The algorithm integrates low-dimensional latent item features, obtained through dimensionality reduction, into a dynamic collaborative filtering framework, enabling the capture of both linear and nonlinear item features. Further, by analyzing user item interactions, it addresses the cold start problem. The inclusion of a neural network attention mechanism refines the algorithm's robustness and efficiency, reducing bias from active users and enhancing recommendation accuracy.



a) HR@10 evaluation metrics for the Pinterest dataset



b) NDCG@10 evaluation metrics for the Pinterest dataset

Figure 8. HR@10 and NDCG@10 metrics for the first 50 iterations of the Pinterest dataset

The proposed algorithm significantly improves traditional recommendation systems, effectively overcoming data sparsity and feature extraction challenges. Moreover, traditional auto-encoder models can handle high-dimensional sparse raw data, but may require increased computational resources when additional data sources are involved. Addressing this, model downsizing or complexity reduction offers a viable solution to balance performance and computational resource requirements.

Acknowledgement

The work was supported by the National Natural Science Foundation of China (No. 62302199), the China Postdoctoral Science Foundation (No. 2023M731368), the

Natural Science Foundation of the Jiangsu Higher Education Institutions (No. 22KJ-B520016), the Jiangsu University Innovative Research Project (No. KYCX223671), the Youth Foundation Project of Humanities and Social Sciences of Ministry of Education in China (No. 22YJC870007), the 2022 Jiangsu University Undergraduate Student English Teaching Excellence Program, and the Ministry of Education's Industry-Education Cooperation Collaborative Education Project (No. 2021023060-05).

REFERENCES

[1] Mumin, D.—Shi, L. L.—Liu, L.—Panneerselvam, J.: Data-Driven Diffusion Recommendation in Online Social Networks for the Internet of People. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 52, 2022, No. 1, pp. 166–178, doi: 10.1109/TSMC.2020.3015355.

[2] Chen, J.—Dong, H.—Wang, X.—Feng, F.—Wang, M.—He, X.: Bias and Debias in Recommender System: A Survey and Future Directions. ACM Transactions on Information Systems, Vol. 41, 2023, No. 3, Art. No. 67, doi: 10.1145/3564284.

[3] Gao, C.—Zheng, Y.—Li, N.—Li, Y.—Qin, Y.—Piao, J.—Quan, Y.—Chang, J.—Jin, D.—He, X.—Li, Y.: A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. ACM Transactions on Recommender Systems, Vol. 1, 2023, No. 1, Art. No. 3, doi: 10.1145/3568022.

[4] Yu, J.—Yin, H.—Xia, X.—Chen, T.—Li, J.—Huang, Z.: Self-Supervised Learning for Recommender Systems: A Survey. IEEE Transactions on Knowledge and Data Engineering, Vol. 36, 2024, No. 1, pp. 335–355, doi: 10.1109/TKDE.2023.3282907.

[5] Shi, L. L.—Liu, L.—Jiang, L.—Zhu, R.—Panneerselvam, J.: QoS Prediction for Smart Service Management and Recommendation Based on the Location of Mobile Users. Neurocomputing, Vol. 471, 2022, pp. 12–20, doi: 10.1016/j.neucom.2021.02.107.

[6] Han, Z.—Shi, L.—Liu, L.—Jiang, L.—Fang, J.—Lin, F.—Zhang, J.—Panneerselvam, J.—Antonopoulos, N.: A Survey on Event Tracking in Social Media Data Streams. Big Data Mining and Analytics, Vol. 7, 2023, No. 1, pp. 217–243, doi: 10.26599/BDMA.2023.9020021.

[7] Shi, L. L.—Liu, L.—Wu, Y.—Jiang, L.—Kazim, M.—Ali, H.—Panneerselvam, J.: Human-Centric Cyber Social Computing Model for Hot-Event Detection and Propagation. IEEE Transactions on Computational Social Systems, Vol. 6, 2019, No. 5, pp. 1042–1050, doi: 10.1109/TCSS.2019.2913783.

[8] Karn, A. L.—Karna, R. K.—Kondamudi, B. R.—Bagale, G.—Pustokhin, D. A.—Pustokhina, I. V.—Sengan, S.: Customer Centric Hybrid Recommendation System for E-Commerce Applications by Integrating Hybrid Sentiment Analysis. Electronic Commerce Research, Vol. 23, 2023, pp. 279–314, doi: 10.1007/s10660-022-09630-z.

[9] Goldberg, D.—Nichols, D.—Oki, B. M.—Terry, D.: Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM, Vol. 35, 1992, No. 12, pp. 61–70, doi: 10.1145/138859.138867.

[10] Jiang, L.—Shi, L.—Liu, L.—Yao, J.—Ali, M. E.: User Interest Community Detection on Social Media Using Collaborative Filtering. Wireless Networks, Vol. 28, 2022, No. 3, pp. 1169–1175, doi: 10.1007/s11276-018-01913-4.

[11] Santos, F. P.—Lelkes, Y.—Levin, S. A.: Link Recommendation Algorithms and Dynamics of Polarization in Online Social Networks. Proceedings of the National Academy of Sciences, Vol. 118, 2021, No. 50, Art. No. e2102141118, doi: 10.1073/pnas.2102141118.

[12] Chen, J.—Zhao, C.—Uliji—Chen, L.: Collaborative Filtering Recommendation Algorithm Based on User Correlation and Evolutionary Clustering. Complex & Intelligent Systems, Vol. 6, 2020, No. 1, pp. 147–156, doi: 10.1007/s40747-019-00123-5.

[13] Deng, J.—Guo, J.—Wang, Y.: A Novel K-Medoids Clustering Recommendation Algorithm Based on Probability Distribution for Collaborative Filtering. Knowledge-Based Systems, Vol. 175, 2019, pp. 96–106, doi: 10.1016/j.knosys.2019.03.009.

[14] Koren, Y.—Bell, R.—Volinsky, C.: Matrix Factorization Techniques for Recommender Systems. Computer, Vol. 42, 2009, No. 8, pp. 30–37, doi: 10.1109/MC.2009.263.

[15] Yu, X.—Jiang, F.—Du, J.—Gong, D.: A Cross-Domain Collaborative Filtering Algorithm with Expanding User and Item Features via the Latent Factor Space of Auxiliary Domains. Pattern Recognition, Vol. 94, 2019, pp. 96–109, doi: 10.1016/j.patcog.2019.05.030.

[16] Yu, X.—Peng, Q.—Xu, L.—Jiang, F.—Du, J.—Gong, D.: A Selective Ensemble Learning Based Two-Sided Cross-Domain Collaborative Filtering Algorithm. Information Processing & Management, Vol. 58, 2021, No. 6, Art. No. 102691, doi: 10.1016/j.ipm.2021.102691.

[17] Peng, Y. X.—Zhu, W. W.—Zhao, Y.—Xu, C. S.—Huang, Q. M.—Lu, H. Q.—Zheng, Q. H.—Huang, T. J.—Gao, W.: Cross-Media Analysis and Reasoning: Advances and Directions. Frontiers of Information Technology & Electronic Engineering, Vol. 18, 2017, No. 1, pp. 44–57, doi: 10.1631/FITEE.1601787.

[18] Silver, D.—Huang, A.—Maddison, C. J.—Guez, A.—Sifre, L. et al.: Mastering the Game of Go with Deep Neural Networks and Tree Search. Nature, Vol. 529, 2016, No. 7587, pp. 484–489, doi: 10.1038/nature16961.

[19] Sun, Q.—Shi, L. L.—Liu, L.—Han, Z. X.—Jiang, L.—Lu, Y.—Panneerselvam, J.: A Dynamic Collaborative Filtering Algorithm Based on Convolutional Neural Networks and Multi-Layer Perceptron. 2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS), 2021, pp. 459–466, doi: 10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00078.

[20] Wu, L.—He, X.—Wang, X.—Zhang, K.—Wang, M.: A Survey on Accuracy-Oriented Neural Recommendation: From Collaborative Filtering to Information-Rich Recommendation. IEEE Transactions on Knowledge and Data Engineering, Vol. 35, 2023, No. 5, pp. 4425–4445, doi: 10.1109/TKDE.2022.3145690.

[21] Han, X.—Wang, Z.—Xu, H. J.: Time-Weighted Collaborative Filtering Algorithm Based on Improved Mini Batch K-Means Clustering. Advances in Science and Technology, Vol. 105, 2021, pp. 309–317, doi: 10.4028/www.scientific.net/AST.105.309.

[22] Cheng, X.—Feng, L.—Gui, Q.: Collaborative Filtering Algorithm Based on Data Mixing and Filtering. International Journal of Performability Engineering, Vol. 15, 2019, No. 8, pp. 2267–2276, doi: 10.23940/ijpe.19.08.p27.22672276.

[23] Bai, H.—Li, X.—He, L.—Jin, L.—Wang, C.—Jiang, Y.: Recommendation Algorithm Based on Probabilistic Matrix Factorization with Adaboost. Computers, Materials & Continua, Vol. 65, 2020, No. 2, pp. 1591–1603, doi: 10.32604/cmc.2020.09981.

[24] Porteous, I.—Asuncion, A.—Welling, M.: Bayesian Matrix Factorization with Side Information and Dirichlet Process Mixtures. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 24, 2010, No. 1, pp. 563–568, doi: 10.1609/aaai.v24i1.7686.

[25] Ahmadian, S.—Ahmadian, M.—Jalili, M.: A Deep Learning Based Trust- and Tag-Aware Recommender System. Neurocomputing, Vol. 488, 2022, pp. 557–571, doi: 10.1016/j.neucom.2021.11.064.

[26] Rostami, M.—Oussalah, M.—Farrahi, V.: A Novel Time-Aware Food Recommender-System Based on Deep Learning and Graph Clustering. IEEE Access, Vol. 10, 2022, pp. 52508–52524, doi: 10.1109/ACCESS.2022.3175317.

[27] Lee, S.—Kim, D.: Deep Learning-Based Recommender System Using Cross Convolutional Filters. Information Sciences, Vol. 592, 2022, pp. 112–122, doi: 10.1016/j.ins.2022.01.033.

[28] Wang, Y.—Zhao, X.—Xu, T.—Wu, X.: AutoField: Automating Feature Selection in Deep Recommender Systems. Proceedings of the ACM Web Conference 2022 (WWW '22), 2022, pp. 1977–1986, doi: 10.1145/3485447.3512071.

[29] Sedhain, S.—Menon, A. K.—Sanner, S.—Xie, L.: AutoRec: Autoencoders Meet Collaborative Filtering. Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion), 2015, pp. 111–112, doi: 10.1145/2740908.2742726.

[30] Strub, F.—Mary, J.—Preux, P.: Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs. NIPS Workshop on Machine Learning for eCommerce, 2015, `https://inria.hal.science/hal-01256422`.

[31] Wu, Y.—DuBois, C.—Zheng, A. X.—Ester, M.: Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM '16), 2016, pp. 153–162, doi: 10.1145/2835776.2835837.

[32] He, X.—Liao, L.—Zhang, H.—Nie, L.—Hu, X.—Chua, T. S.: Neural Collaborative Filtering. Proceedings of the 26th International Conference on World Wide Web (WWW '17), 2017, pp. 173–182, doi: 10.1145/3038912.3052569.

[33] He, X.—Du, X.—Wang, X.—Tian, F.—Tang, J.—Chua, T. S.: Outer Product-Based Neural Collaborative Filtering. CoRR, 2018, doi: 10.48550/arXiv.1808.03912.

[34] Huang, T.—Zhao, R.—Bi, L.—Zhang, D.—Lu, C.: Neural Embedding Sin-

gular Value Decomposition for Collaborative Filtering. IEEE Transactions on Neural Networks and Learning Systems, Vol. 33, 2022, No. 10, pp. 6021–6029, doi: 10.1109/TNNLS.2021.3070853.

[35] Jin, X. B.—Gong, W. T.—Kong, J. L.—Bai, Y. T.—Su, T. L.: PFVAE: A Planar Flow-Based Variational Auto-Encoder Prediction Model for Time Series Data. Mathematics, Vol. 10, 2022, No. 4, Art. No. 610, doi: 10.3390/math10040610.

[36] He, X.—He, Z.—Song, J.—Liu, Z.—Jiang, Y. G.—Chua, T. S.: NAIS: Neural Attentive Item Similarity Model for Recommendation. IEEE Transactions on Knowledge and Data Engineering, Vol. 30, 2018, No. 12, pp. 2354–2366, doi: 10.1109/TKDE.2018.2831682.

[37] Harper, F. M.—Konstan, J. A.: The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems, Vol. 5, 2015, No. 4, Art. No. 19, doi: 10.1145/2827872.

[38] Geng, X.—Zhang, H.—Bian, J.—Chua, T. S.: Learning Image and User Features for Recommendation in Social Networks. 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4274–4282, doi: 10.1109/ICCV.2015.486.

[39] Kabbur, S.—Ning, X.—Karypis, G.: FISM: Factored Item Similarity Models for Top-N Recommender Systems. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13), 2013, pp. 659–667, doi: 10.1145/2487575.2487589.

[40] Airen, S.—Agrawal, J.: Movie Recommender System Using K-Nearest Neighbors' Variants. National Academy Science Letters, Vol. 45, 2022, No. 1, pp. 75–82, doi: 10.1007/s40009-021-01051-0.

[41] Rendle, S.—Krichene, W.—Zhang, L.—Anderson, J.: Neural Collaborative Filtering Vs. Matrix Factorization Revisited. Proceedings of the 14th ACM Conference on Recommender Systems (RecSys '20), 2020, pp. 240–248, doi: 10.1145/3383313.3412488.

[42] Kim, D.—Park, C.—Oh, J.—Lee, S.—Yu, H.: Convolutional Matrix Factorization for Document Context-Aware Recommendation. Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16), 2016, pp. 233–240, doi: 10.1145/2959100.2959165.

[43] Wang, S.—Sun, G.—Li, Y.: SVD++ Recommendation Algorithm Based on Backtracking. Information, Vol. 11, 2020, No. 7, Art. No. 369, doi: 10.3390/info11070369.

Zixuan Han received his B.Sc. degree in 2018, and his M.Sc. degree in 2021, both from the Jilin Normal University, Siping, China. He is currently working towards his Ph.D. degree at the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China. His research interests include data mining, social networks and cloud computing.

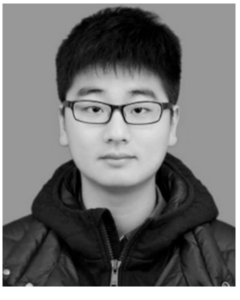
Leilei Shi received his B.Sc. degree from the Nantong University, Nantong, China, in 2012, and his M.Sc. degree in 2015 and his Ph.D. degree in 2020, both from the Jiangsu University, Zhenjiang, China. He is currently a Lecturer at the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China. His research interests include event detection, data mining, social computing and cloud computing.

Qiang Sun received his B.Sc. degree from the Nanjing Sanjiang College, China in 2020. He is currently pursuing his Ph.D. degree at the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China. His research interests include data mining, social networks, link prediction, and recommender systems.

Xiuliang Huang is an undergraduate student at the School of Computer Science and Communication Engineering, Jiangsu University, China. His research interests include data mining, social networks, and cloud computing.

Bing Lei is an undergraduate student at the School of Computer Science and Communication Engineering, Jiangsu University, China. His research interests include deep learning, artificial intelligence, and social networks.

Lu Liu is currently Head of the Department of Electronics, Computing and Mathematics in the University of Derby and Adjunct Professor in the School of Computer Science and Communication Engineering at Jiangsu University. He received his Ph.D. degree from the University of Surrey. He is the Fellow of British Computer Society and Member of IEEE. His research interests are in areas of cloud computing, social computing, data analytics, service-oriented computing and peer-to-peer computing.



Yao Lu is Assistant Professor at the Anhui University, China. Before this, he was a Senior Research Associate at the Lancaster University, UK from 2021 to 2023. He received his Ph.D. degree from the University of Leicester, UK. His current research is focused on energy efficient cloud systems and he has published his recent research works in journals, conferences and as book chapters. His research interests include green cloud computing, big data analytics, and high performance computing. He has won two best paper awards in the IEEE international conferences DSS 2018 and RTDPCC 2020.