

## Research Article

# Collaborative Filtering Recommendation Algorithm for MOOC Resources Based on Deep Learning

Lili Wu 

College of Horticulture and Gardening, Yangtze University, Jingzhou, Hubei 434025, China

Correspondence should be addressed to Lili Wu; [lili.wu@yangtzeu.edu.cn](mailto:lili.wu@yangtzeu.edu.cn)

Received 8 February 2021; Revised 20 March 2021; Accepted 27 March 2021; Published 7 April 2021

Academic Editor: Wei Wang

Copyright © 2021 Lili Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In view of the poor recommendation performance of traditional resource collaborative filtering recommendation algorithms, this article proposes a collaborative filtering recommendation model based on deep learning for art and MOOC resources. This model first uses embedding vectors based on the context of metapaths for learning. Embedding vectors based on the context of metapaths aggregate different metapath information and different MOOCs may have different preferences for different metapaths. Secondly, to capture this preference drift, the model introduces an attention mechanism, which can improve the interpretability of the recommendation results. Then, by introducing the Laplacian matrix into the prior distribution of the hidden factor feature matrix, the relational network information is effectively integrated into the model. Finally, compared with the traditional model using the scoring matrix, the model in this article using text word vectors effectively alleviates the impact of data sparsity and greatly improves the accuracy of prediction. After analyzing the experimental results, compared with other algorithms, the resource collaborative filtering recommendation model proposed in this article has achieved better recommendation results, with good stability and scalability.

## 1. Introduction

At present, many universities comply with the indicators of the Ministry of Education, using the advantages of the Internet, artificial intelligence, big data analysis technology, putting forward intelligent education. In the implementation of smart education, MOOC resources are the most important component [1–3]. Therefore, the design of a collaborative filtering recommendation algorithm for MOOC resources has very important research significance [4, 5].

With the continuous development of big data technology, MOOC resource recommendation algorithms have emerged. With the help of recommendation technologies in the fields of e-commerce, tourism routes, and social networks, many excellent MOOC resource recommendation algorithms have been proposed [6, 7]. Traditional courses only have dozens or hundreds of students, but a MOOC can hold more than 100,000 people at most. Therefore, in the Internet, collaborative filtering is used to recommend art learning MOOC resources [8, 9]. Due to the increasing

volume of resource data, the existing collaborative filtering recommendation algorithm for art learning MOOC resources can only stay on the surface of the data, resulting in higher MAE values. Deep learning can automatically extract deep features. Therefore, in order to solve the current problems in the process of collaborative filtering recommendation for MOOC resources, this article designs a collaborative filtering recommendation algorithm for MOOC resources based on deep learning.

The remainder of this article is organized as follows. Section 2 reviews the related work. Section 3 introduces the proposed methods. Section 4 reports the experiments and results. Section 5 concludes our work.

## 2. Related Works

Nowadays, the recommendation system has been widely used in all occupations, and it is unknowingly changing people's lives. According to statistics from foreign media VentureBeat, Amazon increased its revenue by about 35% in

2003, all thanks to its introduction of an item-based collaborative filtering recommendation algorithm [10–13]. Traditional collaborative filtering methods use shallow machine learning models, unable to learn the deep characteristics of users and items. Therefore, the fusion of side information for recommendation is getting more and more attention, and hybrid recommendation is becoming more and more popular. However, auxiliary information often has problems such as large scale, multiple types, inconsistent data types, and missing key data. The hybrid recommendation is facing severe challenges [14, 15].

In recent years, deep learning has made huge breakthroughs in many directions of artificial intelligence, and its application in the direction of recommendation systems is still in its infancy. The recommender system branch of the International Computer Society held a symposium on the application of deep learning in the recommendation field and pointed out in the meeting that the next important direction of recommender system research would be deep learning [16]. Therefore, it is of great significance to apply deep learning to the research of recommender systems [17].

The algorithm based on nearest neighbors mainly uses the method of calculating the similarity of users or items in the scoring matrix to make recommendations for users. Many scholars have improved and innovated this algorithm. A typical representative is the item-based algorithm [18], which has been successfully applied to Amazon's e-commerce system. Wang et al. [19] proposed a collaborative filtering recommendation algorithm based on matrix factorization, which has greatly improved the accuracy of recommendation. Subsequently, scholars continued to improve it and successively proposed the probabilistic matrix factorization (PMF) model [20], the SVD++ model [21], the factorization machines model [22], and so forth and achieved certain improvements to varying degrees. Hazrati et al. [23] proposed a collaborative filtering algorithm based on restricted Boltzmann machines (RBM). For the first time, deep learning is introduced into the recommendation system to learn the hidden factors of users and items. Later, some scholars made improvements on its basis, but the RBM-based collaborative filtering algorithm has many drawbacks, such as long training time and large scale of weight parameters connecting the hidden layer and the visible layer, which is difficult to implement in practical applications. Chen et al. [24] applied a deep belief network (DBN) to the recommendation system and proposed a new deep hybrid recommendation model. Before the introduction of deep learning, Liu et al. [25] proposed a topic model (CTR). It models the abstract of the article and then learns the hidden feature representation of the article, which is used for article recommendation. However, when the auxiliary information is very sparse, the hidden features obtained by the topic model for learning are insufficient and invalid. At this time, it is the turn of deep learning to show its effect. Huang et al. [26] directly used CNN and DBNs to obtain hidden factors from content information. However, it only considers the hidden factors of the item and only applies to music data. Zhao et al. [27] proposed a collaborative deep learning (CDL) model to obtain hidden features from the

textual information of the project. This model uses the SDAE [28] instead of LDA, which solves the problem of insufficient learning of hidden features when the data are sparse in collaborative topic regression.

The emergence of the CDL model instantly attracted a lot of attention and became a benchmark for a large number of researchers to improve and compare. In the CDL model, the author expressed the auxiliary information of the article through the bag-of-words (BoW) model. This model uses unordered words to express text and cannot dig out important information implicit in the order of words. Therefore, Zhang et al. [29] used the idea of a cyclic neural network [30] in the encoding and decoding stage to improve the SDAE model into a collaborative recurrent autoencoder, which makes the correlation information between words in the auxiliary information dug out. Li et al. [31] proposed a collaborative variational autoencoder (CVAE) for the noise reduction extension SDAE. The model does not need to add noise to the input; it can better learn the hidden factors of the item from the auxiliary information. Since CDL only uses the superficial meaning of auxiliary information, Xiong et al. [32] introduced CNN in the text recommendation to mine the relevant information within the auxiliary information. They used word-embedding technology to replace the BoW model to represent words in auxiliary information. Then, these words were concatenated to form a two-dimensional matrix. Then, convolution, pooling, and fully connected mapping were performed on this two-dimensional matrix to finally obtain the hidden factor of the item.

Under the trend of deep learning, vertical and horizontal recommendation, in which a deep learning model is adopted to mine hidden features and a collaborative filtering algorithm is adopted to combine them with different recommendation scenarios, is a trend and direction [33–36].

### 3. Abnormal Behaviour Detection Algorithm

*3.1. Spark Architecture.* The advantage of the Spark architecture is that it is more suitable for data-mining algorithms, which can search for hidden information from large amounts of data. The Spark framework includes functional components such as SQL queries, text processing, and machine learning. These components are tightly integrated within Spark. The computational performance is better, especially in environments where information is analyzed and iterated in large quantities, and the advantages of using the Spark framework are particularly great. So this article chose the Spark architecture.

With the widespread popularity of MOOCs, a large number of MOOC resources can be searched on many Internet social platforms. If you want to search for the information you need in massive resources, you need to use the collaborative filtering recommendation algorithm under the Spark architecture, which is achieved by establishing the user's own search history model, recording the user's preferences and interests, and actively providing relevant MOOC push. From the point of view of the collaborative filtering algorithm, similar users will have similar tastes. Therefore, the preferences of similar users can be used to

make recommendations for the target users. The algorithm flow is shown in Figure 1.

The Spark framework includes functional components such as SQL query, text processing, and machine learning. These components are tightly integrated into Spark. Its computing performance is better, especially in the work environment of mass analysis and iteration of information; the advantages of using the Spark framework are more prominent. When the Spark architecture is running in a cluster, the driver first completes the resource application through the resource manager. After the manager allocates resources, the Executor is started on the corresponding node. After the node completes the task submitted by the driver, it finally sends the request to the driver. The program submits feedback.

**3.2. MOOC Resource Score Prediction Based on Dual-Channel CNN.** The input of the recommendation system based on deep learning is generally the relevant information between the MOOC classes, and the deep neural network model will automatically learn the implicit representations between the MOOC classes and recommend courses for users based on the Cain representation. A basic deep learning recommendation system includes an input layer, a modeling layer, and an output layer. In the modeling layer, most of the deep learning models used include RBM, convolutional neural networks, and recurrent neural networks. In the output layer, the deep network model learns a highly abstract representation between MOOCs and then generates a project recommendation list through steps such as inner product, Softmax conversion into probability values, and similarity calculation and ranking.

The traditional matrix decomposition method only uses the scoring information, so it only learns the representation of the MOOC. In addition to scoring information, this model also uses other additional information. Therefore, in addition to the representation learning between classes, this model also explicitly learns the metapath-based context representation between the user and the course. As can be seen in Figure 2, this model is mainly composed of two modules, one of which is embedding vector learning based on the context of metapaths. The second module is to introduce the Laplacian matrix into the prior distribution of the hidden factor feature matrix, and the relational network information is effectively integrated into the model.

Path instances of different metapaths are input to the hierarchical neural network to learn their low-dimensional vector representation. The path instance vectors of all specific metapaths are pooled to obtain the low-dimensional vector representation of the metapath itself. The context based on metapaths aggregates different metapath information, and different MOOCs may have different preferences for different metapaths. In order to capture this preference drift, the model introduces an attention mechanism. Attention mechanism has been widely used in the field of natural language processing to learn the importance of different words or sentences. The introduction of an

attention mechanism can not only produce better performance but also improve the interpretability of recommendation results.

The model in this article can effectively integrate user scoring information, the content of the MOOC, and network information. In addition, the characteristic expressions learned can be circulated among the MOOC's relationship network so that the characteristic expressions can be more precise and can more accurately describe the implicit characteristic vectors in the MOOC. By introducing the Laplace matrix into the prior distribution of the social hidden factor matrix, the relational network information is effectively integrated into the model.

It can be seen from the generation process of the model that the model in this article successfully integrates the course feature expression vector obtained through deep learning, the user's rating matrix  $\beta$ , and the network matrix  $\alpha$ , so that the feature representation is more accurate and the MOOC can be described more accurately. The recommended framework of this model is shown in Figure 3. First, it is necessary to preprocess the training dataset, convert the user's collection records of the courses into the user-rating matrix, S\_matrix, and use the BoW model to express the title and summary information of the MOOC as the content matrix, C\_matrix, and the reference relationship between the MOOCs as social adjacency matrix, J\_matrix. Then, these three kinds of information are fused, and two neural networks with the same input and different output are trained simultaneously. The user implicit eigenmatrix and MOOC implicit eigenmatrix are obtained, and the predicted results are finally obtained.

The current calculation formulas for the similarity between MOOC resource A and MOOC resource B mainly include cosine similarity, Pearson correlation coefficient, and constrained Pearson correlation coefficient. Their calculation formulas are from formulas (1)–(3). The selection formula (3) of the MOOC resource collaborative filtering recommendation algorithm proposed in this article calculates the similarity between MOOC resource A and MOOC resource B.

$$S(A, B) = \frac{\sum_{i=1}^n F_{A_i} \times F_{B_i}}{\sqrt{\sum_{i=1}^n F_{A_i}^2} \sqrt{\sum_{i=1}^n F_{B_i}^2}} \quad (1)$$

$$\text{Person}(A, B) = \frac{\sum_{i=1}^n (F_{A_i} - F_A) \times (F_{B_i} - F_B)}{\sqrt{\sum_{i=1}^n (F_{A_i} - F_A)^2} \sqrt{\sum_{i=1}^n (F_{B_i} - F_B)^2}} \quad (2)$$

$$C\_Person(A, B) = \frac{\sum_{i=1}^n \alpha(F_{A_i} - F_A) \times \beta(F_{B_i} - F_B)}{\sqrt{\sum_{i=1}^n (F_{A_i} - F_A)^2} \sqrt{\sum_{i=1}^n (F_{B_i} - F_B)^2}} \quad (3)$$

Among them,  $F_A$  represents the predicted feature score of MOOC resource A, and  $n$  represents the number of MOOC resources.

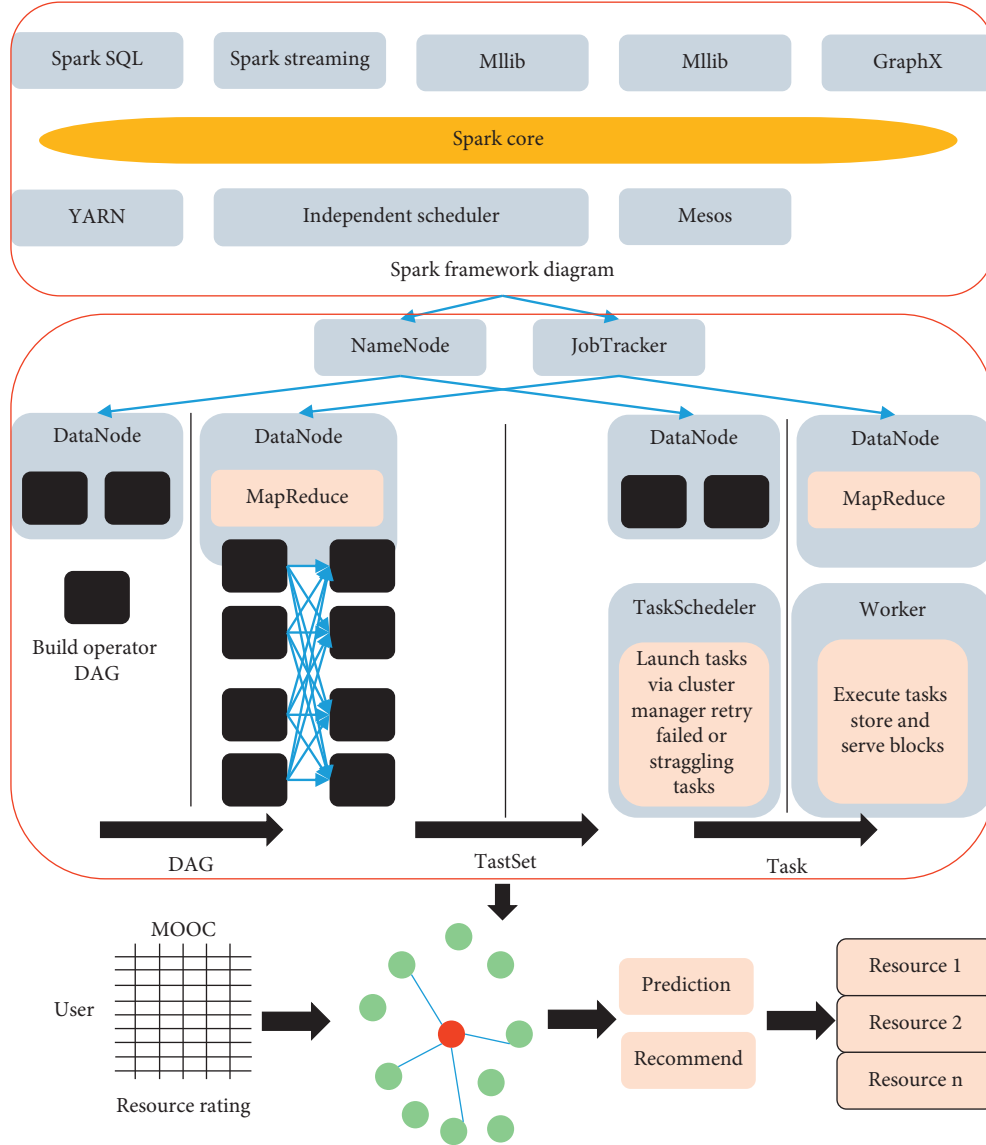


FIGURE 1: Schematic diagram of the collaborative filtering recommendation algorithm flow based on Spark.

**3.3. MOOC Recommendation Algorithm Based on Word-Embedding Vector.** In recent years, word-embedding vectors have been widely used in many applications of natural language processing, making the training of the model of an end-to-end overall process instead of a traditional pipeline. It does not rely on feature engineering and greatly improves the performance of the system. Through the word-embedding model, the long text is mapped to another space by function  $G$ ; namely,  $G: W \rightarrow W_m$ , where  $W$  is a dictionary composed of words in the review text or description text and  $W_m$  is the  $m$ -dimensional vector mapped by the function  $G$ . This article uses this representation technology to mine the semantics of the review text and MOOC description text. In the input layer of the model, the review text and MOOC description text are, respectively, represented as a matrix of word-embedding

vectors so that their semantic information can be learned. Specifically, all comments of user  $a$  are divided into a single document  $w$ , which contains a total of  $n$  words. Then, a word vector matrix  $E$  is constructed for user  $a$ ; the rule is as follows:

$$E_{1:n}^a = \psi(w_1^a) \otimes \dots \otimes \psi(w_n^a). \quad (4)$$

Among them, variable  $w_1^a$  represents the first word in the document  $w$ . Through matrix  $E$ , the order of words can be maintained.

After a series of operations on the CNN layer, you can learn the hidden features of users and MOOCs. However, these two features come from the comment content and the description text, respectively, and are not in the same feature space, so it is impossible to perform factorization and other operations.

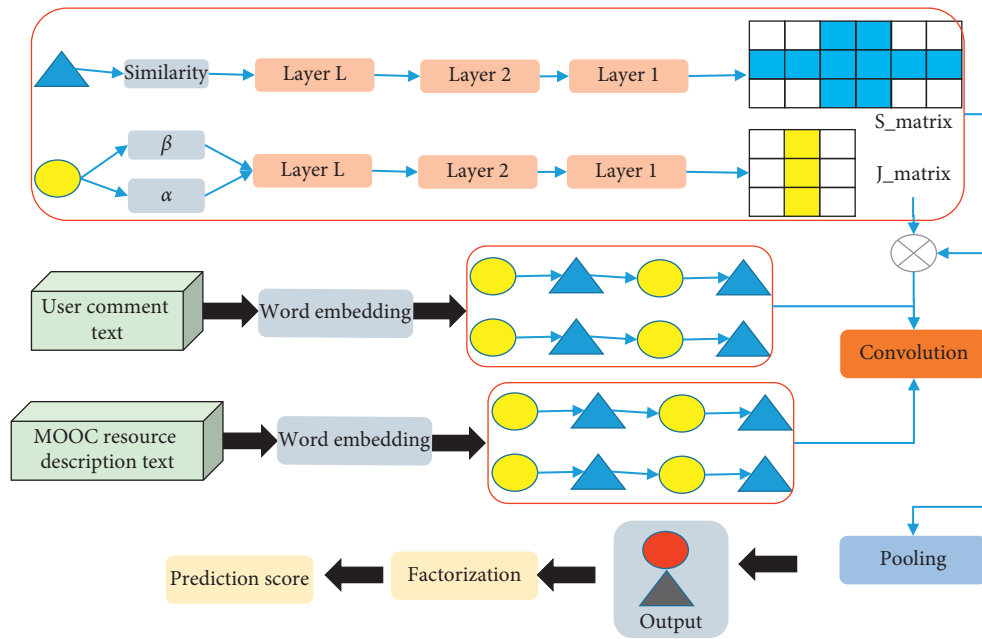


FIGURE 2: Algorithm framework diagram based on deep learning.

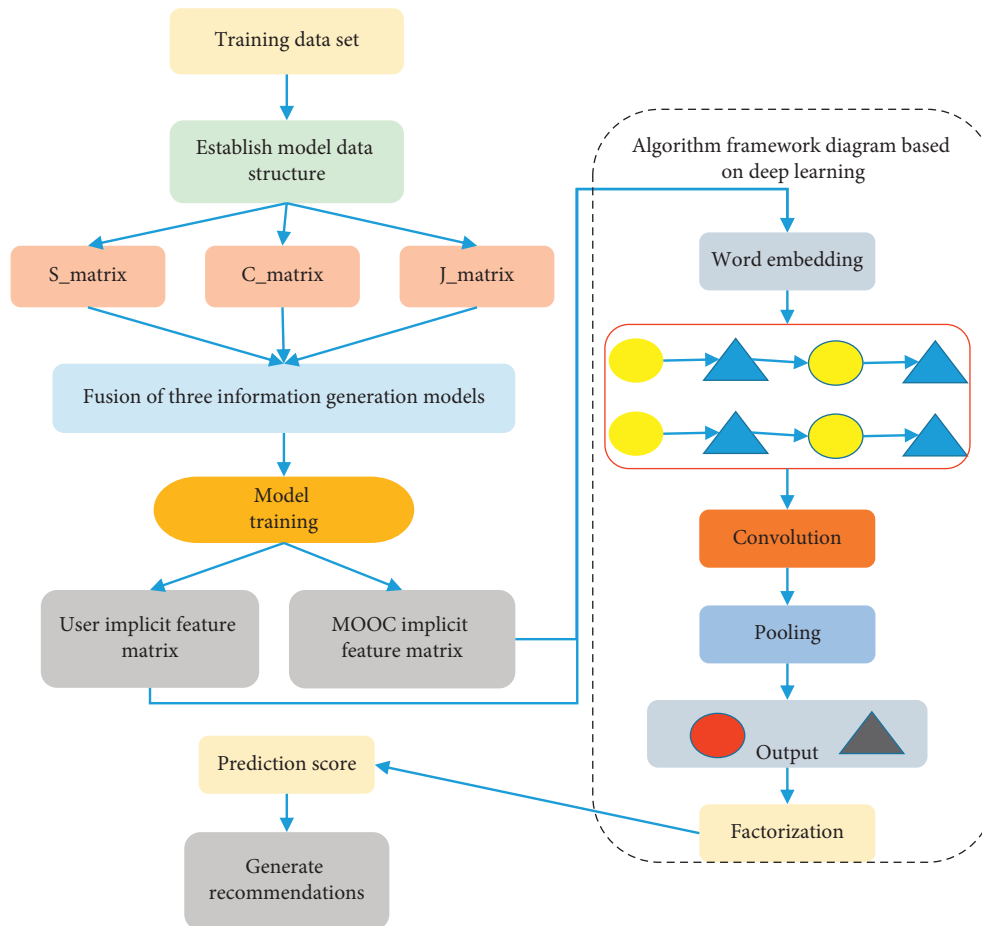


FIGURE 3: Framework diagram of collaborative filtering recommendation based on deep learning.

Therefore, it is necessary to use a shared structure first and merge the two into the same feature space before subsequent processing can be performed and the shared layer has emerged.

First, a separate correlation vector  $\varepsilon = (\varepsilon_1, \varepsilon_2)$  needs to be constructed to connect the user implicit feature  $\varepsilon_1$  output by the CNN model with the MOOC implicit feature  $\varepsilon_2$ . Then, the model-based hidden factor model is used in the recommendation system to model the association vector  $u$  and train the final predictive score. For a given training sample, the loss function is shown in

$$\text{Loss} = \text{actual} - (\varepsilon_0 + \varepsilon_1 u_1 + \varepsilon_2 u_2). \quad (5)$$

Among them, the variable actual is the actual value of the score, variable  $\varepsilon_0$  is the overall bias term of the entire model, and variable  $\varepsilon_1$  is the weight of vector  $u_1$ .

## 4. Results and Discussion

**4.1. Datasets.** A large number of MOOC resource experiment objects are selected, and they are divided into ten categories. The number of MOOC resources in each category is shown in Table 1.

**4.2. Experimental Parameter Settings.** In addition to MOOC content information, the model can also make recommendations using relational network information. If the scoring parameters  $\alpha$  and  $\beta$  are set to 1, it means that this model only uses MOOC scoring information or only integrates network information for recommendation. For other values of  $\beta$  and  $\alpha$ , it means that the model proposed in this article combines scoring information and network content information at the same time.

Figure 4(a) shows the effect of different values of parameter  $\alpha$  on the recall rate when  $\beta$  is fixed. This article sets  $\beta = 10$ . It can be seen from the figure that the value of  $\beta$  is relatively sensitive to the impact of recommended performance. As the value of  $\alpha$  increases, the recommendation performance also gradually increases. When the value of  $\alpha$  is 10, the model in this article achieved the highest recall rate and then began to decline. When the value of  $\alpha$  exceeds 10 or more, the recommended performance is significantly worse. The reason is that the excessive alpha value causes the MOOC resources in the mutual relationship to be too close to each other, thus making the prediction results false.

Figure 4(b) shows the effect of parameter  $\beta$  on the recall rate when  $\alpha$  is fixed. This article sets  $\alpha = 20$ . It can be seen from the figure that the value of  $\beta$  is not sensitive to the impact of recommended performance. As  $\beta$  value increases, the recommendation performance also slowly improves. When the value of  $\beta$  exceeds 20, the recommended performance begins to decline slowly. This is because, for very small  $\beta$ , the model in this article is approximately equivalent to CDL. As the value of  $\beta$  increases, the model in this article also incorporates more network information to improve recommendation performance. An excessively large  $\beta$  value indicates that the model has a serious tendency to relational network information.

The parameters in Figure 5 are all trained ten times, and each training uses a fivefold cross-validation method. The

TABLE 1: Quantity distribution of MOOC resources.

Number	MOOC resource name	Quantity
1	Language	2000
2	Legal	1500
3	Biological	1000
4	Computer	1000
5	Art	1000
6	Building	300
7	Logistics	200
8	Mathematics	1500
9	Materials science	500
10	Chemistry	800

average MSE corresponds to the average MSE of the ten experiments and the number of iterations is the number of iterations required for the current model loss value to stabilize. It can be seen from Figure 5(a) that the best effect can be achieved when the word vector dimension is 100. The reason is that as the dimension of the word vector increases, the word vector in the high-dimensional space changes from dense to sparse, which weakens the association between words. In Figure 5(b), the number of convolution kernels from one to five layers is 16. Studies have shown that the best results can be achieved by using a 3-layer convolution module. The reason is that the number of convolution parameters used in the first layer is the largest, the number of dimensionality reductions is small, the output matrix is large, the text feature extraction is not sufficient, the final fully connected layer has many neural units, and the training speed is slow. Under the 5-layer convolution module, the features extracted by the convolution module are too abstract. It can be seen from Figure 5(b) that the effect of using a 3-layer convolution module is the best, so the comparison experiments with different numbers of convolution kernels here use a layer convolution structure. From Figure 5(c), the optimal number of convolution kernels is 16. The analysis found that using four and eight convolution kernels, the abstract feature ability is limited, and the convolution kernel is not fully used to extract the deep features between the user and the MOOC. When using 32 and 64 convolution kernels, the extracted features are too detailed and noise is abstracted, resulting in overfitting and long training time.

**4.3. The Impact of Similarity Function and Negative Sample Ratio on Recommendation Performance.** The similarity function can be used to measure the degree of similarity between MOOC classes. Figure 6 compares the impact of different similarity functions on the recall rate. Cos means using cosine similarity as a similarity function. Pearson means using the Pearson correlation coefficient as the similarity function. Constraint Pearson means using the constrained Pearson correlation coefficient as the similarity function. It can be seen from the figure that the recommended performance of the model using the similarity function constraint Pearson is the best. It can be seen that the similarity function will affect the recommendation performance, and constraint Pearson has the best performance.

The objective function of the recommendation system can be divided into two categories: point-by-point and pairwise.

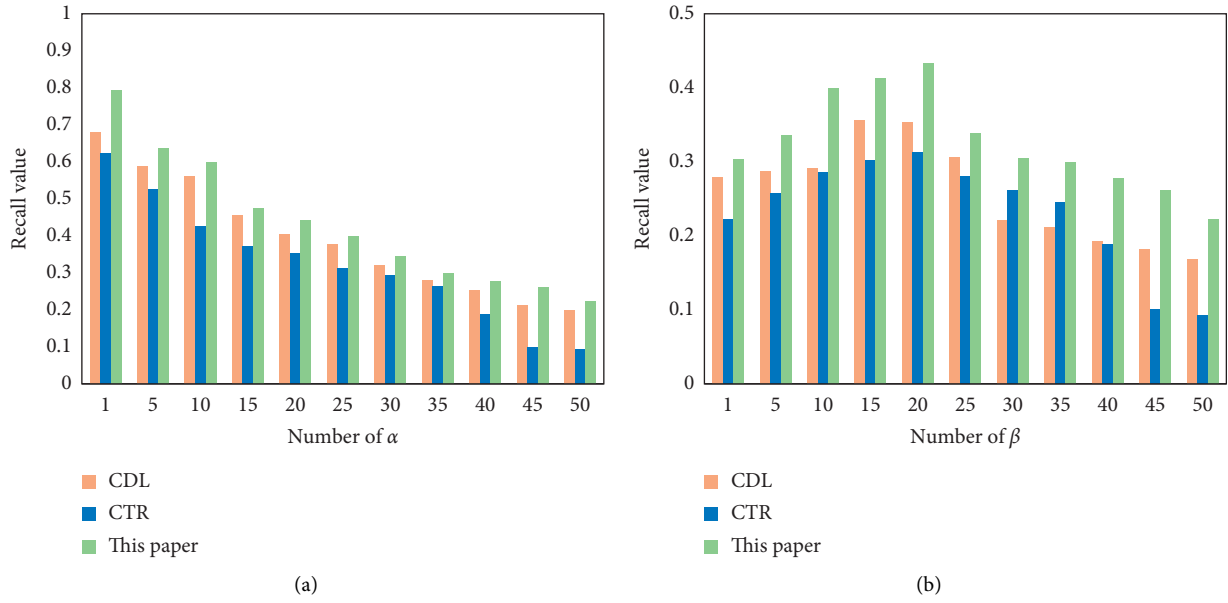


FIGURE 4: The effect of  $\alpha$  and  $\beta$  on recommended performance. (a) Impact of  $\alpha$  on recommended performance. (b) Impact of  $\beta$  on recommended performance.

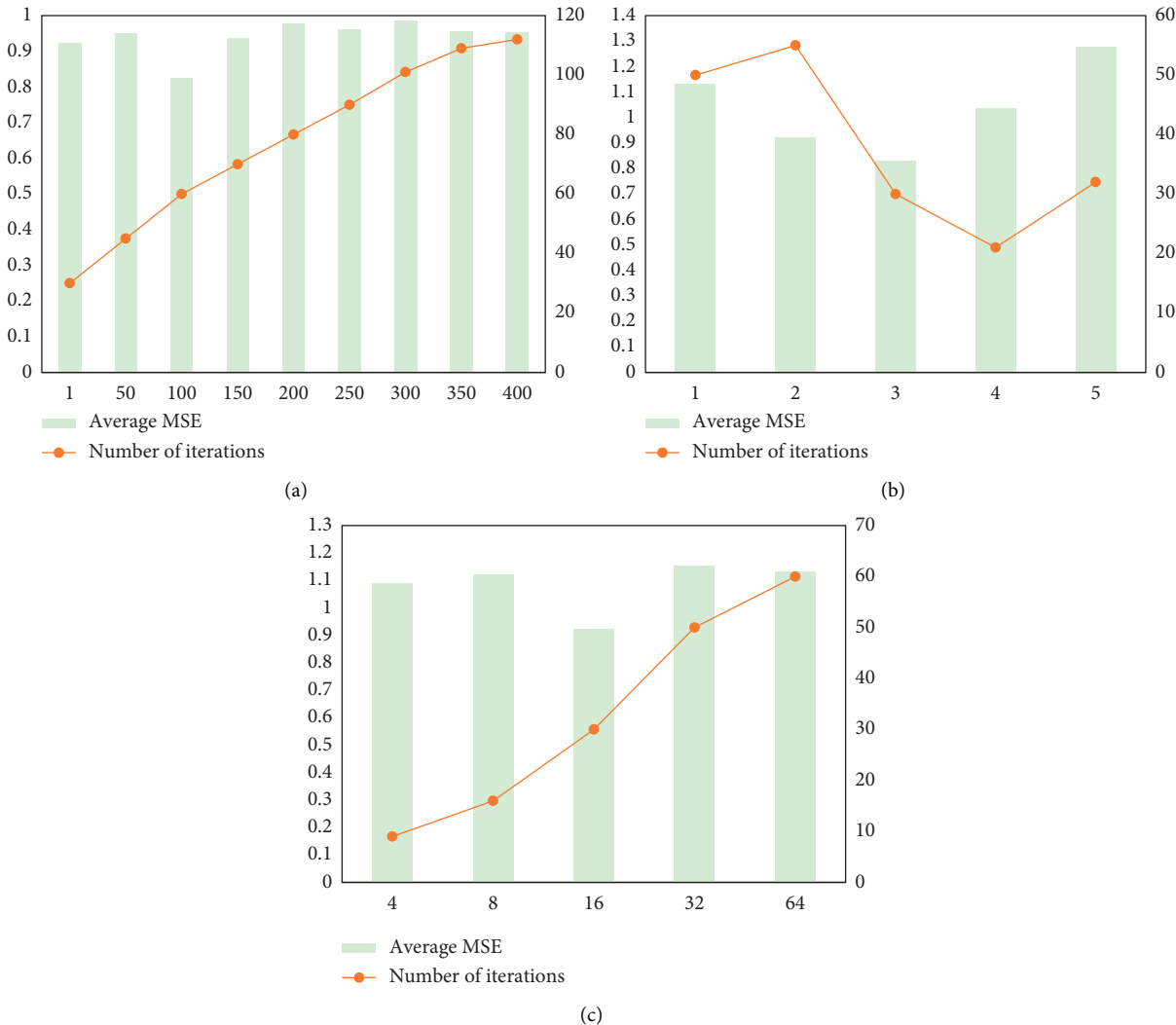


FIGURE 5: The influence of different numbers of parameters on the model in this article. (a) Comparison of related indicators under different word vector dimensions. (b) Comparison of related indicators under different convolutional layers. (c) Comparison of related indicators under different numbers of convolutional kernels.

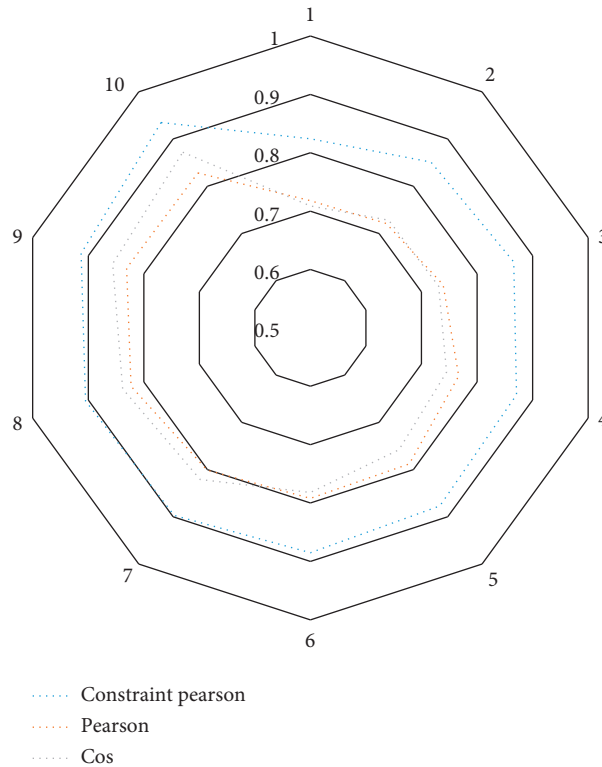


FIGURE 6: The impact of different similarity functions on recommendation performance.

Compared with the paired objective function, the point-by-point objective function is more free to choose the proportion of negative samples. In order to clarify the influence of the negative sample ratio on the recommended performance, this article does the following experiment to capture the relationship between the negative sample ratio and performance.

Figure 7 shows the performance of the model with different negative sample ratios under different datasets and different indicators. It can be seen from the figure that when the proportion of negative samples is below four, the performance will be greatly improved as the proportion of negative samples increases. However, when the proportion of negative samples exceeds four, as the proportion of negative samples increases, the performance will increase, but the magnitude is relatively small. In addition, the dataset size is  $N$  times the positive sample, where  $N$  is the proportion of negative samples plus one. Therefore, when the positive samples are large, the proportion of negative samples with large sampling will cause the training time to increase exponentially. In order to balance performance and time complexity, choosing a small negative sample ratio will be the best choice. Therefore, the best choice for a negative sample ratio is between six and eight.

**4.4. Comparison and Analysis of the Accuracy of MOOC Resource Recommendation.** In the above experimental environment parameters, the algorithm based on the cloud platform, the traditional algorithm based on shallow

machine learning, and the algorithm in this article are, respectively, used to conduct experiments. The experimental results are shown in Table 2.

The experimental results show that, with the decrease of training set and the increase of test set, the MAE and RMSE values of the three algorithms are all decreasing, and the accuracy is constantly improving. The MAE and RMSE values of the algorithm proposed in this article are lower than the other two algorithms in each proportion, and the accuracy is higher than the other two algorithms. The traditional algorithm adopts the shallow machine learning model, unable to learn the deep features of users and items. This indicates that the Spark-based collaborative filtering recommendation algorithm for MOOC resources of art has higher recommendation accuracy and better performance.

In order to verify the stability of the model in this article, the dataset was divided into fivefold cross-validation in the experiment, and 50 rounds of experiments were conducted. The results are shown in Figure 8. The loss value in the 50-round experiment is negatively correlated with the number of training rounds, there will be no large-scale jitter in the overall training process, and the lowest learning rate can be reached with a large learning rate. Experimental results show that the model in this article has good stability.

Select the resource collaborative filtering recommendation algorithms of literature [19], literature [23], and literature [27] to conduct comparative experiments. For the same dataset, their MOOC resource recommendation accuracy is shown in Figure 9.



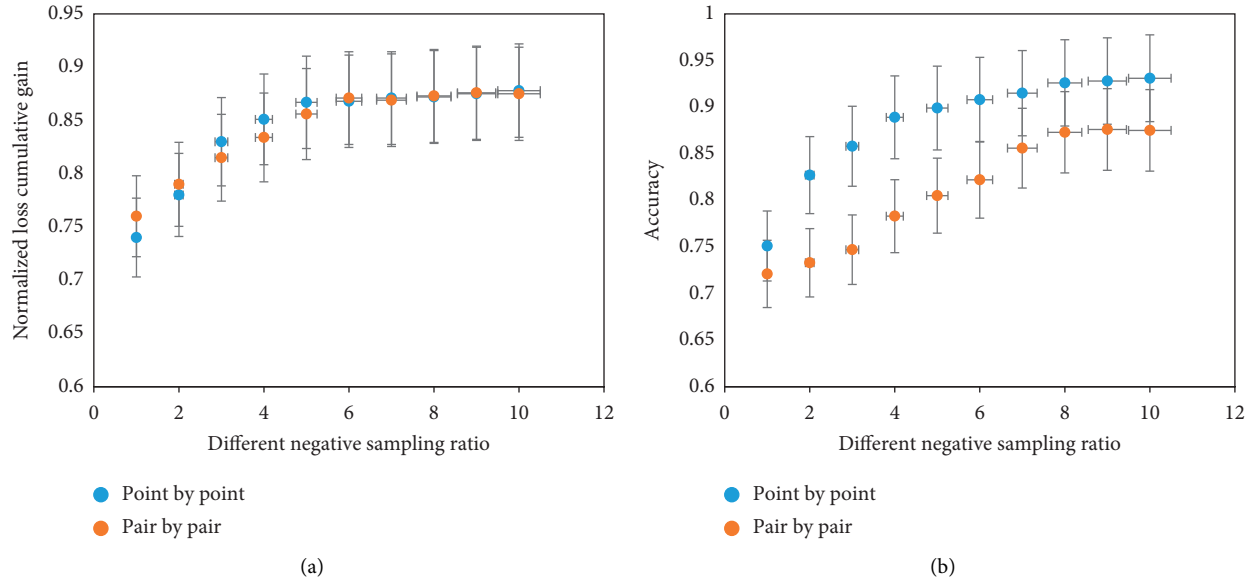


FIGURE 7: Loss gain and accuracy under different negative sampling ratios. (a) Normalized loss cumulative gain under different negative sampling ratios. (b) Accuracy under different negative sampling ratios.

TABLE 2: MAE, RMSE, and accuracy of different proportions of the training set and test set.

Training set: test set	Algorithm based on cloud platform			Based on traditional algorithms			Algorithm based on Spark		
	Accuracy	RMSE	MAE	Accuracy	RMSE	MAE	Accuracy	RMSE	MAE
9:1	0.762	0.749	0.562	0.799	0.775	0.601	0.661	0.672	0.452
8:2	0.789	0.729	0.532	0.795	0.763	0.583	0.691	0.633	0.401
7:3	0.795	0.707	0.501	0.819	0.729	0.532	0.735	0.623	0.389
6:4	0.811	0.689	0.476	0.827	0.707	0.501	0.789	0.566	0.321

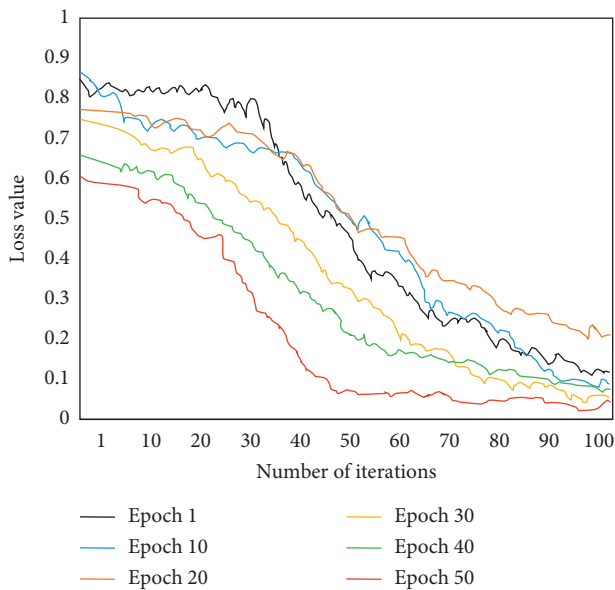


FIGURE 8: The training process of this model.

A comparative analysis of the recommendation accuracy of MOOC resources in Figure 9 shows that the accuracy of MOOC resource recommendation in this model is much

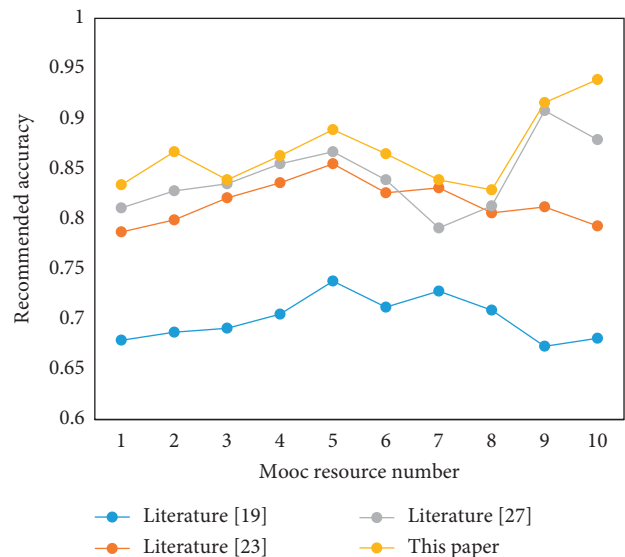


FIGURE 9: Comparison of recommendation accuracy of MOOC resources with different algorithms.

higher than that of literature [19], literature [23], and literature [27]. It reduces the error of MOOC resource recommendation.

## 5. Conclusion

With the continuous development of big data technology, MOOC resource recommendation algorithms have emerged. The system analyses students' learning interests based on the students' learning history and related materials. Foreign countries pay more attention to the research of MOOC resource recommendation. They have proposed many excellent MOOC resource recommendation algorithms with the help of recommendation technologies in the fields of e-commerce, tourism routes, and social networks. However, the current recommendation performance of these algorithms is still poor. Therefore, based on the Spark architecture, this article proposes a collaborative filtering recommendation model based on deep learning for art education resources. This model is mainly composed of two modules, one of which is embedding vector learning based on the context of metapaths. The second module is to introduce the Laplacian matrix into the prior distribution of the hidden factor feature matrix, and the relational network information is effectively integrated into the model. Compared with the traditional model using the scoring matrix, the model using the text word vector effectively alleviates the impact of data sparsity and greatly improves the accuracy of prediction. After analyzing the experimental results, compared with other algorithms, the resource collaborative filtering recommendation model proposed in this article has achieved better recommendation results, with good stability and scalability.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The author declares no conflicts of interest.

## References

- [1] S. Tang, "Learning Mechanism and Function characteristics of MOOC in the process of higher education," *Eurasia Journal of Mathematics, Science and Technology Education*, vol. 13, no. 12, pp. 8067–8072, 2017.
- [2] E. Ayub and L. C. Leong, "Developing a pedagogy framework for institution-wide implementation of MOOC: a case study from a Malaysian private university," *Advanced Science Letters*, vol. 23, no. 2, pp. 809–813, 2017.
- [3] P. Guo, "MOOC and SPOC, which one is better?," *Eurasia Journal of Mathematics, Science and Technology Education*, vol. 13, no. 8, pp. 5961–5967, 2017.
- [4] H. Zhang, T. Huang, Z. Lv, S. Liu, and H. Yang, "MOOCRC: a highly accurate resource recommendation model for use in MOOC environments," *Mobile Networks and Applications*, vol. 24, no. 1, pp. 34–46, 2019.
- [5] K. Mingsiritham and G. Chanyawudhiwan, "Experiment of the prototype of online learning resources on massive open online course (MOOC) to develop life skills in using technology media for hearing impaired students," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, no. 03, pp. 242–249, 2020.
- [6] H. Zhang, T. Huang, Z. Lv, S. Liu, and Z. Zhou, "MCRS: a course recommendation system for MOOCs," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 7051–7069, 2018.
- [7] A. Khalid, K. Lundqvist, and A. Yates, "Recommender systems for MOOCs: a systematic literature survey," *The International Review of Research in Open and Distributed Learning*, vol. 21, no. 4, pp. 255–291, 2020.
- [8] J. Shu, X. Shen, H. Liu, B. Yi, and Z. Zhang, "A content-based recommendation algorithm for learning resources," *Multimedia Systems*, vol. 24, no. 2, pp. 163–173, 2018.
- [9] J. Xiao, M. Wang, B. Jiang, and J. Li, "A personalized recommendation system with combinational algorithm for online learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 3, pp. 667–677, 2018.
- [10] E. U. Okon, B. O. Eke, and P. O. Asagba, "An improved online book recommender system using collaborative filtering algorithm," *International Journal of Computer Applications*, vol. 179, no. 46, pp. 41–48, 2018.
- [11] Y. Jdidou and M. Khaldi, "Using recommendation systems in MOOC: an innovation in education that increases the profitability of students," in *Enhancing Knowledge Discovery and Innovation in the Digital Era*, M. D. Lytras, L. Daniela, and A. Visvizi, Eds., p. 15, IGI Global, Harrisburg, PA, USA, 2018.
- [12] L. Xiaojun, "An improved clustering-based collaborative filtering recommendation algorithm," *Cluster Computing*, vol. 20, no. 2, pp. 1281–1288, 2017.
- [13] P. Zhang, Z. Zhang, T. Tian, and Y. Wang, "Collaborative filtering recommendation algorithm integrating time windows and rating predictions," *Applied Intelligence*, vol. 49, no. 8, pp. 3146–3157, 2019.
- [14] X. Liu, "A collaborative filtering recommendation algorithm based on the influence sets of e-learning group's behavior," *Cluster Computing*, vol. 22, no. 2, pp. 2823–2833, 2019.
- [15] J. Deng, J. Guo, and Y. Wang, "A Novel K-medoids clustering recommendation algorithm based on probability distribution for collaborative filtering," *Knowledge-Based Systems*, vol. 175, pp. 96–106, 2019.
- [16] L. Zhang, T. Luo, F. Zhang, and Y. Wu, "A recommendation model based on deep neural network," *IEEE Access*, vol. 6, pp. 9454–9463, 2018.
- [17] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, vol. 69, pp. 29–39, 2017.
- [18] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-n recommendation," *ACM Transactions on Information Systems*, vol. 37, no. 3, pp. 1–25, 2019.
- [19] Y. Wang, Z. Song, and C. Xiao, "Collaborative filtering recommendation algorithm based on improved clustering and matrix factorization," *Journal of Computer Applications*, vol. 38, no. 04, pp. 1001–1006, 2018.
- [20] Y. Yin, L. Chen, y. xu, and J. Wan, "Location-aware service recommendation with enhanced probabilistic matrix factorization," *IEEE Access*, vol. 6, pp. 62815–62825, 2018.
- [21] J. Jiao, X. Zhang, F. Li et al., "A novel learning rate function and its application on the SVD++ recommendation algorithm," *IEEE Access*, vol. 8, pp. 14112–14122, 2019.
- [22] K. Kitai, J. Guo, S. Ju et al., "Designing metamaterials with quantum annealing and factorization machines," *Physical Review Research*, vol. 2, no. 1, Article ID 013319, 2020.

- [23] N. Hazrati, B. Shams, and S. Haratizadeh, "Entity representation for pairwise collaborative ranking using restricted Boltzmann machine," *Expert Systems with Applications*, vol. 116, pp. 161–171, 2019.
- [24] R.-C. Chen, "User rating classification via deep belief network learning and sentiment analysis," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 3, pp. 535–546, 2019.
- [25] C. Liu, T. Jin, S. C. H. Hoi, P. Zhao, and J. Sun, "Collaborative topic regression for online recommender systems: an online and Bayesian approach," *Machine Learning*, vol. 106, no. 5, pp. 651–670, 2017.
- [26] Z. Huang, G. Shan, J. Cheng et al., "TRec: an efficient recommendation system for hunting passengers with deep neural networks," *Neural Computing and Applications*, vol. 31, no. 1, pp. 209–222, 2019.
- [27] L. Zhao, Q. Wang, Q. Zou et al., "Privacy-preserving collaborative deep learning with unreliable participants," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1486–1500, 2019.
- [28] X. Yan, Y. Liu, and M. Jia, "Health condition identification for rolling bearing using a multi-domain indicator-based optimized stacked denoising autoencoder," *Structural Health Monitoring*, vol. 19, no. 5, pp. 1602–1626, 2020.
- [29] G. Zhang, Y. Liu, and X. Jin, "A survey of autoencoder-based recommender systems," *Frontiers of Computer Science*, vol. 14, no. 2, pp. 430–450, 2020.
- [30] J. Bueno, S. Maktoobi, L. Froehly et al., "Reinforcement learning in a large-scale photonic recurrent neural network," *Optica*, vol. 5, no. 6, pp. 756–760, 2018.
- [31] X. Li and J. She, "Collaborative variational autoencoder for recommender systems," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 305–314, Halifax, NS, Canada, August 2017.
- [32] M. T. Xiong, Y. Feng, T. Wu, J. X. Shang, B. H. Qiang, and Y. N. Wang, "TDCTFIC: a novel recommendation framework fusing temporal dynamics, CNN-based text features and item correlation," *IEICE Transactions on Information and Systems*, vol. 102, no. 8, pp. 1517–1525, 2019.
- [33] F. Yang, H. Wang, and J. Fu, "Improvement of recommendation algorithm based on collaborative deep learning and its parallelization on Spark," *Journal of Parallel and Distributed Computing*, vol. 148, pp. 58–68, 2021.
- [34] S. H. Ahmed, V. H. C. De Albuquerque, and W. Wei, "Guest editorial: special section on advanced deep learning algorithms for industrial Internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2764–2766, 2021.
- [35] W. Wang, X. Zhao, Z. Gong, Z. Chen, N. Zhang, and W. Wei, "An attention-based deep learning framework for trip destination prediction of sharing bike," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 3, pp. 535–546, 2020.
- [36] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, "Accurate and fast URL phishing detector: a convolutional neural network approach," *Computer Networks*, vol. 178, Article ID 107275, 2020.