Hindawi
Scientific Programming
Volume 2022, Article ID 5448442, 8 pages
https://doi.org/10.1155/2022/5448442

Research Article

Research on Recommendation of Big Data for Higher Education Based on Deep Learning

Ang Zhao[1] and Yanhua Ma[2]

[1]*Dean's Office, Nanjing University of Economics and Finance, Nanjing 210023, China*
[2]*School of Mechano-Electronic Engineering, Hebei Normal University of Science & Technology, Qinhuangdao 066600, China*

Correspondence should be addressed to Yanhua Ma; mayanhua3557@hevttc.edu.cn

Received 18 January 2022; Accepted 14 March 2022; Published 19 May 2022

Academic Editor: Man Fai Leung

To improve the recommendation accuracy of educational resources, an intelligent recommendation method based on autoencoder has been proposed by combining intelligent recommendation algorithm and autoencoder. The method uses the dimension reduction advantage of autoencoder to obtain the required feature vector. Then, the prediction score is utilized to recommend educational resources. Finally, it is verified from the algorithmic and system perspective. The results show that this recommendation method is the most efficient method. The efficiency on the dataset is 0.90, respectively. Furthermore, it can score different recommended articles, and the recommendation of different educational resources is realized.

1. Introduction

For the improvement of users' attention, most e-commerce platforms such as Taobao and JD have developed intelligent recommendation systems based on big data technology, which has attracted a large number of users for them. An intelligent recommendation system is designed to attract customer traffic. However, with the blessing of front-end technologies such as Internet of things, mobile Internet, and so on, the intelligent recommendation system shows a broader application space, thus attracting the attention of many scholars. One of the research hotspots is to introduce big data recommendation system in the field of higher education. According to students' big data in the educational scene, specific education information, education resources, and others can be pushed to specific students, which can improve the utilization rate of educational resources and students' learning efficiency. Regarding the study of recommendation algorithms, Duan created the collaborative filter recommendation algorithm, which takes both the direct impact of expert users on prediction scores and the indirect influence of trustees on prediction scores into account [1]. Thus, it shows better application performance. Ya-Zhi et al. created an adaptive learning service recommendation algorithm based on big data, which has excellent performance in coverage, accuracy, recall rates, and others. In addition, the algorithm is especially useful for learning service recommendation practices [2]. Han et al. improved the traditional collaborative filtering algorithm and proposed a time-weighted collaborative filtering algorithm based on the clustering of mini batch K means. Compared with the traditional algorithm, the accuracy of rating prediction has been significantly improved. Also, the application space of the collaborative filtering algorithm has been further expanded [3]. In addition, Huang et al. created diversified recommendation algorithms for specific application scenarios. The autoencoder based on the deep learning algorithm can achieve feature learning, data dimension reduction, and other functions [4]. Introducing autoencoder into the recommendation system can obtain more ideal application effects. For example, Simpson et al. introduced a stacked noise reduction autoencoder in the recommendation system and supplemented by project information and user information, and thus the quality of its recommendation has been greatly improved [5]. Wu et al. also proposed the big data recommendation method for educational resources and verified the effect of recommendation methods, respectively [6–10]. Therefore, based on the above research, a recommendation method of big data for higher education resources is proposed, and the feasibility of this method is verified. However, due to the huge number of higher education data resources, the film viewing data of middle school students on campus network are taken as the entry point to explore the recommendation of higher education big data resources.

2. Convolutional Denoising Autoencoder

Autoencoder is essentially a kind of unsupervised network, which can be divided into three different parts, namely, encoder, decoder, and hidden layer. The functions of each part are different. The encoder and decoder are mainly used to realize the conversion of data between different dimensions. The specific structure is shown in Figure 1 [11, 12].

In this network, the relationship between the hidden layer and the input layer can be expressed as follows:

$$y = S(W_1 x + b_1). \tag{1}$$

It can be seen from the above formula that x and y represent the data of input and hidden layer, respectively. b_1 and W_1 represent the bias and weight of adjacent nodes, respectively. $s(x)$ represents the corresponding activation function. This formula actually corresponds to the encoding process, while the decoding is expressed as follows:

$$z = S(W_2 y + b_2), \tag{2}$$

where z represents the data of output layer; $\frac{2}{b}$ and W_2 represent the bias and weight of adjacent nodes, respectively; and $s(y)$ represents the corresponding activation function.

Based on AE and adding some noise, the denoising autoencoder can be obtained. The advantage is that it can get the characteristics of high robustness. The specific structure is shown in Figure 2 [13].

In this structure, $L(\overline{x}, z)$ represents the loss function, which generally needs to be placed at a lower level to maintain a high consistency between the features obtained by the hidden layer and the original data. The basic form of the function is as follows:

$$L(\overline{x}, z) = -\sum_{n=1}^{d} \left[x_n 1 g z_n + (1 - \overline{x}_n) 1 g (1 - z_n)\right]. \tag{3}$$

The above analysis shows that the process of adding noise actually is to process the data of two neurons (1, 3). The reconstructed data $\overline{x}$ can be obtained after setting to 0; then, the difference value between the input and output data can be calculated, so as to realize the update of offset vector and weighting matrix, which ensures that the model can be trained continuously.

3. Collaborative Filtering Recommendation of Hybrid Self-Coding Network Model

Based on the previous analysis, the basic definition of self-coding model is defined, and on this basis, the recommendation algorithm is studied and designed. The algorithm is mainly divided into two steps, and the first is the process of obtaining feature vectors. In this stage, the denoising autoencoder is used to obtain the required feature vectors [14]. The second is to predict the score, which needs to make use of the feature vectors obtained in the previous step to learn the interactive network model, and then the scoring results can be obtained. The basic structure of the network is shown in Figure 3.

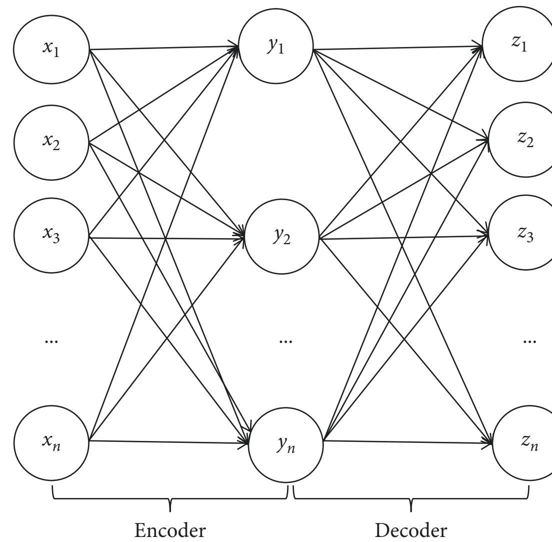

Figure 1: The network structure of autoencoder.

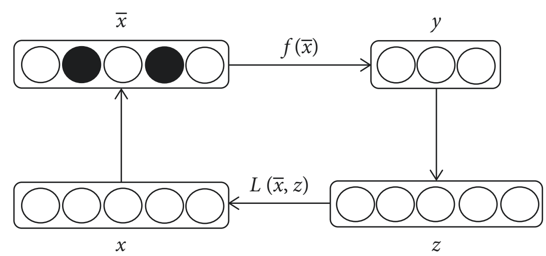

Figure 2: The network structure of denoising autoencoder.

The score depends on the students' interest level in the article, and the overall score ranges from 1 to 5. The highest and the lowest interest scores are 5 points and 1 point, respectively, which means that with the increase of the score, the interest degree gradually increases.

The number of articles and users is n and m, respectively, and the set of the two is represented as $I = \{I_1, I_2, \ldots, I_n\}$ and $U = \{U_1, U_2, \ldots, U_m\}$ in turn. It can be seen that the corresponding interaction scoring matrix is $R_{m \times n}$. Assuming that there are user u and article i, the score that u gives i is expressed as $R_{u,i}$. If it is equal to zero, it does not mean that the user does not like the article but may not have read the article and therefore did not score it.

In practice, there may be multiple users scoring the same article or one user scoring multiple articles. Among them, the vector obtained after different users scoring the same article i is $(R_{1i}, R_{2i}, \ldots, R_{ni}) \in R_m$. Also, the vector formed after a user scoring each article is $(R_{1i}, R_{2i}, \ldots, R_{ni}) \in R_m$.

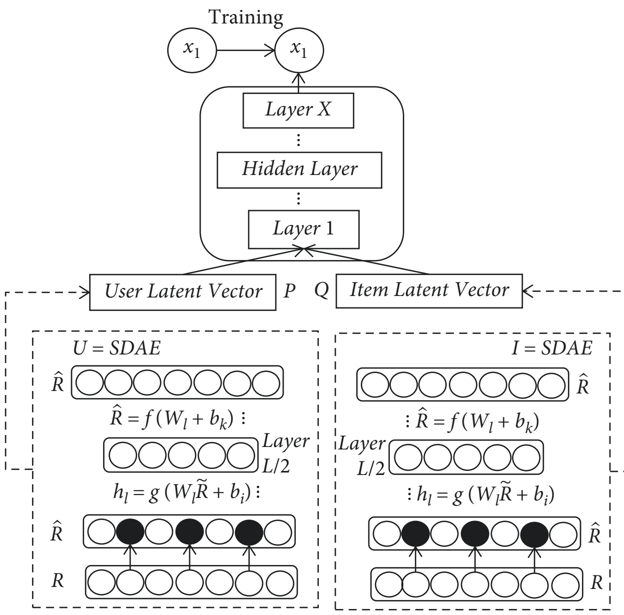

FIGURE 3: The network model diagram of hybrid autoencoder.

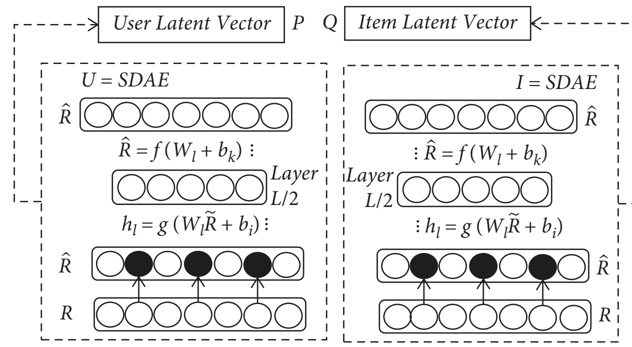

FIGURE 4: The network model diagram of constructing latent feature vector.

3.1. User-Item Latent Vector Construction. As shown in Figure 4, the left and right sides are the feature vectors of learning users and learning articles, which are represented as $U-\mathrm{SDAE}$ and $I-\mathrm{SDAE}$, respectively. They are basically consistent in the structure [15, 16].

The latter is taken as an example for analysis in this study, and the specific contents are shown below.

In the research process, the stack autoencoder structure is adopted. There are some differences between the network and the traditional autoencoder, and the number of the hidden layers is more. The input data are mainly $R = \{r_1, r_2, \ldots, r\}_n, r(u) = (R_{u1}, R_{u2}, \ldots, R_{un})$.

First of all, the damage score matrix $\tilde{R} = (\tilde{r}_1, \tilde{r}_2, \tilde{r}_3, \ldots, \tilde{r}_n)$ is obtained, that is, random noise is added to the original data. Then, the matrix is connected to the autoencoder network, where the low-dimensional feature vector Q can be obtained through coding, and then R_i can be obtained through decoding.

The basic form of each hidden layer is as follows [17]:

$$h_l = g(W_l\tilde{R} + b_l). \tag{4}$$

The output layer L is expressed as

$$\tilde{R} = f(W_l + b_R). \tag{5}$$

The target loss function is shown as follows:

$$\min \|\tilde{R} - R\|. \tag{6}$$

Among them, f and g represent the encoding and decoding functions in turn, and the encoding and decoding parts are the front and back $l/2$ layers in the network, respectively. In addition, the activation function adopted in the training is sigmoid function, and the objective function is used to reduce the reconstruction error and keep it within the appropriate range.

After training, the corresponding latent feature vectors, including article and user vectors, can be obtained. At the same time, compared with the input data, the two dimensions are significantly reduced, which is mainly related to the application of autoencoder [18, 19]. It can be seen that it avoids the occurrence of sparsity. Then, add the obtained latent feature vectors to the interaction model and continue to perform the subsequent processing.

3.2. Learning User-Item Interaction Networks. By establishing the user-item interaction network, the deep information between the item and the user can be obtained and applied to the subsequent prediction. Based on the above process, the project and user feature vector matrix can be obtained. Q and P are used as the embedding layers of the network.

The predicted result can be obtained after passing multiple hidden layers. Each hidden layer here actually belongs to a deep semantic relationship. Also, the predicted result y can be obtained in the process of reducing their dimension. Significantly, the data that are not scored should be eliminated in the training, that is, the scored data must be used.

The deviation between the target value and the predicted value should be controlled at a low level in the process of training, and the objective function should be determined according to this principle. As can be seen from the above analysis, the basic form of the interactive network is obtained as follows [20]:

$$\tilde{y}_{ui} = f\left(P^k, Q^k | P, Q, \theta_f\right). \tag{7}$$

Here, $f(\beta)$ represents the interaction function, θ_f represents the corresponding model parameters, θ_f and $P \in R_{m\times k}$ represent the potential feature vector matrices of users and items, respectively, and the basic form of loss function is shown as follows [21–23]:

$$L_{sqr} = \sum_{(u,i)\in Z} \left(y_{ui} + \tilde{y}_{ui}\right)^2. \tag{8}$$

The above formula shows that Z represents the scored data, which needs to be used in the training. However, there may be missing values, which need to be processed by certain methods. Otherwise, the accuracy of prediction results will be inevitably reduced. Some scholars have proposed different processing methods, and the commonly used method is to calculate the average score. Although missing values can be processed, there is still shortcoming, which is that different users adopt different scoring methods. So, the scored data are directly used in the training process. In the training, the algorithm is optimized based on the stochastic gradient descent method, and the prediction score is obtained. The specific form is as follows [24]:

$$\widetilde{R}_{ui} = f\left(P^u, Q^i | P, Q, \theta_f\right), \quad (9)$$

where Q^i and P^u correspond to the latent feature vectors of i and u, respectively, and the predicted results can be obtained after processing based on interactive network.

The implementation of the collaborative filtering recommendation algorithm based on hybrid autoencoder has been described in detail. Also, this algorithm is systematically referenced in this paper. First of all, it is necessary to preprocess the collected 201028 user scoring data. The collected data include the article data filtering out the rating behavior less than 20 times and the user data filtering out the interaction behavior less than 20 times. Thus, 186532 user scoring data are obtained, which are included in the experimental training set. The training parameters of the network are as follows: the noise rate of the stack noise reduction autoencoder is 0.2. The activation function is the sigmoid function, and the dimension of implicit feature vector is $k \in 10, 40, 80, 200, 400$. The number of network layer neurons in U-SDAE module is 128-64-32-64-128 in turn. In addition, the learning rate is 0.005, and the dropout is 0.15. The number of network layer neurons in I-SDAE module is 128-64-32-64-128 in turn. The learning rate is 0.005, and dropout is 0.15. Furthermore, the learning rate of deep interactive neural network is 0.001. The activation function is sigmoid function, and dropout is 0.10. Here, stitching project and user's hidden semantic vectors are used to construct the embedding layer of interactive neural network. The number of network layer neurons is 64-32-8 in order.

After completing the parameter setting, the prediction score of the user u for the article i can be obtained through the algorithm operation, which helps the article to be sorted. Also, the Top-N article is selected to be included in the user recommendation list.

Use the content-based recommendation algorithm to determine the Top-N recommendation list of courses and articles and use the collaborative filtering recommendation algorithm to determine the Top-N recommendation list of another article. Thus, a total of one course Top-N recommendation list and two article Top-N recommendation lists are obtained. Also, the two article Top-N recommendation lists are re-sorted in accordance with the established rules, so as to obtain the unique version of the article Top-N recommendation list. The calculation process is as follows.

The article Top-N lists determined by content-based recommendation algorithms and based on collaborative filtering recommendation algorithms are B and F, respectively, and the final version of article Top-N list is R. The equation is as follows [25]:

$$R = B \cap F + \alpha B + \beta F. \quad (10)$$

The weight $a = 0.6$ and $\beta = 0.4$ are obtained through experimental training, and the final version of the article Top-N recommendation list is determined to achieve the recommendation service.

4. Verification of Recommendation Algorithm

4.1. Verification of Algorithm Effect. The interactive behavior data between the item and the user are analyzed and applied to the designed recommendation algorithm; meanwhile, the use of the latent factor mode is helpful to solve the sparse problem. Then, it is necessary to test and analyze the application effect of the model, and appropriate datasets must be selected. In this design, taking the video data that students browse on the campus network of some colleges and universities, dataset A analyzing learning video that can be used for big data analysis is constructed, and then the big data recommendation method constructed in this study is utilized to make recommendations.

The basic information for the dataset is shown in Table 1.

The data should be uniformly divided into two parts: training set for the training process and test set for the test process. Also, the number of the two should be reasonably set. The ratio of the two is set as 9 : 1. Appropriate indicators are used to evaluate and analyze the application effect of the recommendation algorithm, and RMSE is a root sign based on MAE, which can better describe the error of data. Therefore, RMSE is selected in this paper.

In addition, the denoising autoencoder adopts sigmoid function, where the noise rate is 0.3, and the latent feature vector dimension is $k \in 20, 40, 80, 200, 400, 500$. The basic parameters of each module are set. Here, the number of neurons in module $U-$SDAE is 943-700-400-700-943, and the number of neurons in module $I-$SDAE is 1682-900-400-900-1682. The learning rate of the two modules is consistent with that of dropout, which are 0.004 and 0.15, respectively. For the interaction network part, dropout and learning rate are 0.15 and 0.001, respectively. Selecting the sigmoid function, the corresponding number of neurons is 800-400-32.

Comparing and analyzing the application effect of the algorithm and using the quantitative indicators to evaluate, the difference in application effect between the algorithm and other models is analyzed. For UserAverage and Item-Aaverage, they both use the mean scores of items and users. For SVD algorithm, it is necessary to analyze whether there are missing data first. If there is a need to be filled, the mean score is adopted in this process. The K dimension after decomposed should meet certain requirements, namely, sum of the squares of the first k singular values needs to achieve 90% of the total singular value, and then the similarity of different users should be calculated. On this basis, the final score can be obtained. For autoencoder, the number of neurons is 943-500-943, and the unobserved data, hidden layer dimension, and regular coefficient are 3, 500, and 0.001,

Table 1: Training dataset.

Number of users	Number of items	Number of interaction records	Sparsity
6040	3706	More than 1 million	2.57%

respectively. For PMF, stochastic gradient descent and exponential decay methods are used in the process of model training and learning rate setting, respectively, so as to realize the improvement of the model.

According to the obtained feature matrix, the score prediction results can be obtained. The efficiency information of each model is shown in Table 2.

Table 2: Efficiency information of each model.

Model	Dataset of students watching a video recording
UserAverage	0.76
ItemAaverage	0.83
SVD	0.85
PMF	0.86
Autoencoder	0.87
Ours	0.90

4.2. Practical Application Verification of the Algorithm

4.2.1. System Architecture Design. The platform for learning career development is researched and designed, and the intelligent recommendation algorithm is integrated in the whole system, which is convenient to combine users' needs and preferences to recommend interested information, including book information, course information, and so on. The core part of the system is the collaborative filtering algorithm, which combines with the actual teaching requirements to design the system function modules, so as to meet the personalized needs of students in learning. Combined with the basic requirements of the whole platform to design the system architecture scheme, the specific implementation route is shown in Figure 5 [26].

As can be seen from the figure, the whole system is generally divided into three layers, namely, data layer, application layer, and recommendation layer. Each layer is connected, which realizes the overall function based on data sharing and interaction mode. The basic introduction of each part is as follows.

(1) Data Layer. The data layer is the basic part of the whole platform. It mainly realizes the storage and management of basic data and responds to data requests from other parts. After completing the operation, the relevant data information can be returned. In addition to the management of basic business data, the generated recommendation result data are also involved. The common data management mode is relational database, which needs to design the standardized data tables for management. Considering the basic requirements of system performance, in addition to using the traditional relational database, the non-relational database is used in this design, which helps to achieve higher response speed. The crawler tool can efficiently obtain the educational information from the network and then save it in the database, which can be used for the subsequent queries and processing operations.

(2) Application Layer. This layer belongs to the core part of the whole platform, which needs to realize the basic logical business and complete the implementation of each functional module based on the user's needs. Considering the needs of system expansion and upgrade, the whole is divided into two parts, namely, front-end and back-end. The former is mainly related to interface display and layout. The latter mainly realizes the specific logical functions and data transmission, and based on the coordination of the two, it can present the required pages and data for the users. This part is divided into several modules, such as online communication and course reservation, and so on. It needs to call the relevant interface of WeChat server to complete the authorization operation and obtain the user's profile picture, nickname information, and other information. After the authorization is successful, other functions can be used.

(3) Recommendation Layer. This part realizes the recommendation function, and it combines the collected basic data for unified processing and analysis, including business system data and user behavior data. In addition, this module will form personalized recommendation for different users' needs and interests, so as to meet different users' requirements. In this part, the Spark framework is used to improve the processing efficiency with the help of the distributed processing platform. Meanwhile, the user interest model needs to be updated regularly, and the collected log data are utilized to improve the service quality.

4.2.2. Implementation of the Recommendation Service Based on Spark. The previous analyses show that the basic principle of recommendation service has been clarified. This part will design the basic structure and process of the whole recommendation framework, which is specifically divided into multiple processes, such as data source, data processing, and so on. The basic implementation route is shown in Figure 6 [27].

In the above framework design, considering the amount of higher education resources and the scalability of the system, the Spark framework is adopted to construct the system, and HDFS is utilized to store massive higher education resource data, so as to better lay the foundation for subsequent big data analysis.

4.2.3. System Development Environment. According to the configuration of the basic environment of system development and combined with the previous analysis, it can be seen that the system is generally divided into two parts,

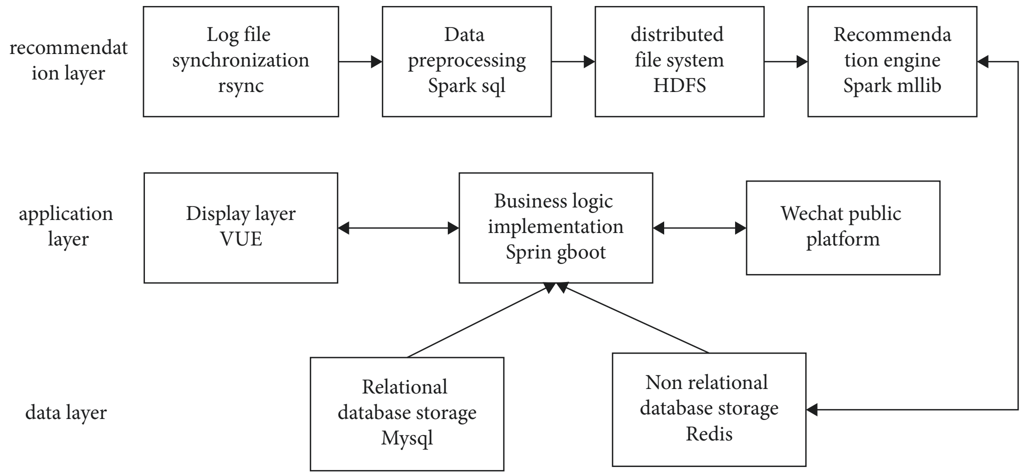

Figure 5: Implementation route of the system architecture scheme.

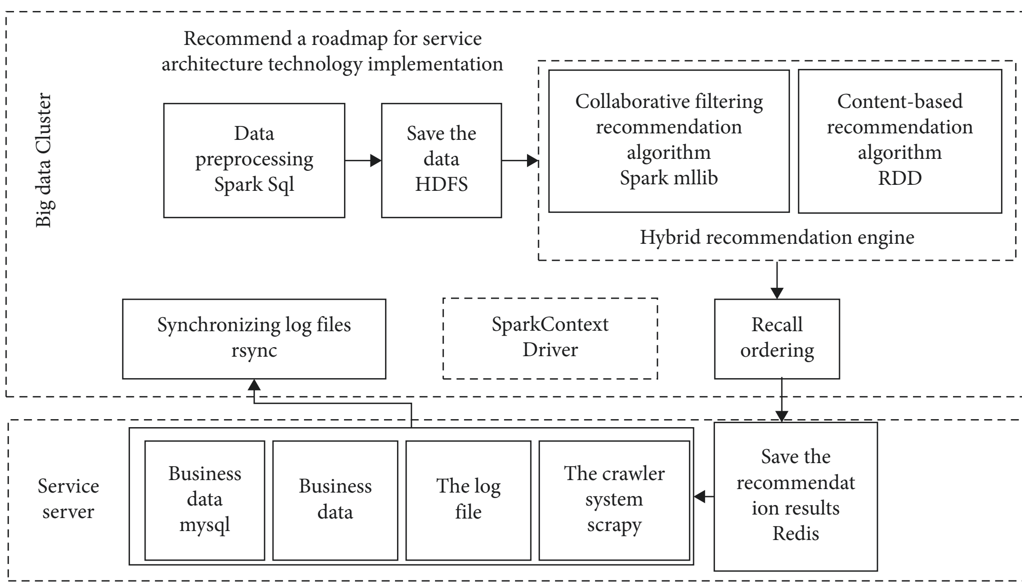

Figure 6: Implementation route of the recommendation service architecture scheme.

Table 3: Big data cluster configuration information.

The serial number	Hardware and software information	Configuration information
1	CPU	8 cores and 16 threads
2	Memory	16 G
3	The operating system	CentOS 6.8
4	Scala version	2.12.0
5	Hadoop version	2.7.3
5	Spark version	2.1.0
6	Maven version	3.3.9

among which the front-end part adopts WebStorm and Vue. The background uses MySQL, IntelliJ IDEA, and Spring Boot. In addition to the above parts, the mature distributed frameworks including Spark and others are used in big data processing.

Three servers are used to configure the Spark cluster. The specific parameters are given in Table 3.

The installation process is divided into several steps, which is shown as follows.

(1) Download the Spark and JDK installation packages from the network.
(2) Install and configure JDK.
(3) Configure SSH to deploy the cluster.

Table 4: Big data cluster machine parameter.

Serial number	Machine name	IP address
1	Hadoop001	58.1 19.1 12.15
2	Hadoop002	58.1 19.112.16
3	Hadoop003	58.1 19.112.17

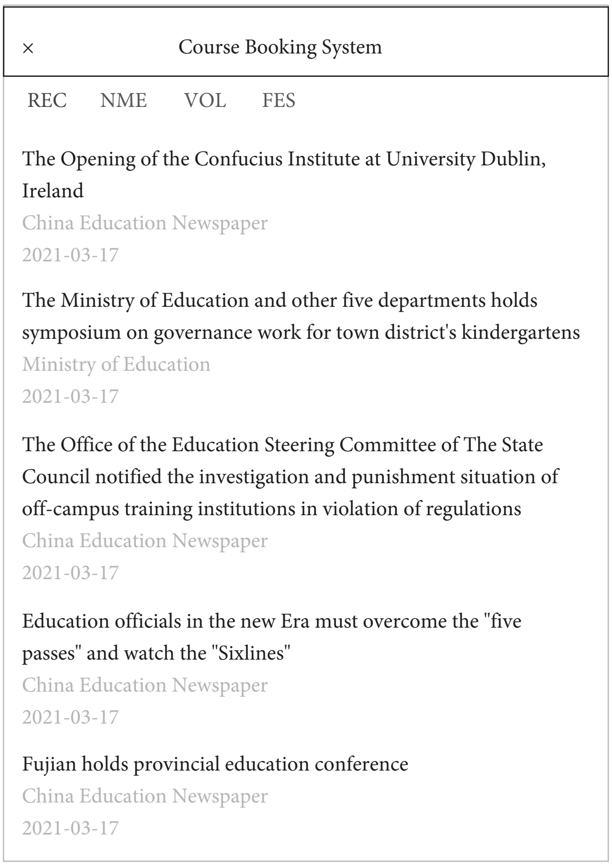

Figure 7: Recommendation result interface.

(4) After the above steps, install and configure Hadoop and set the information in the configuration file.

(5) Install and configure Spark by referring to Step (4). The basic parameters are given in Table 4.

The Jenkins scheme is adopted in this design, which can automatically and efficiently complete packaging, deployment, and other operations. Users can execute Maven operation after submitting push operation, which helps the jar packaging operation to be realized. In consideration of the possible errors in the above process, an effective notification mechanism is designed, that is, the relevant information is timely transmitted to the developer through e-mail, which ensures the accuracy and reliability of the deployment process.

4.2.4. Recommendation Results. The core part of the recommendation system is the collaborative filtering recommendation algorithm, which needs to calculate the item and user feature vectors first. Here, the RDD operation is mainly utilized, and the recommendation results can be obtained after the calculation of interactive similarity. Spark MLlib is used in the algorithm implementation, but the neural network class needs to be designed by oneself. After the algorithm design is completed, inputting data can get the recommendation result, and then save the final result in Redis. The recommendation result interface is shown in Figure 7.

5. Conclusion

The recommendation system in this paper effectively overcomes the sparsity problem of the recommendation algorithm. It performs well in precision, coverage, and other aspects, which can meet the personalized recommendation needs of educational resources. The recommendation system improves the traditional feature processing method, and neural network is used to extract feature vector, which greatly improves the recommendation accuracy.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Z. Duan, W. Xu, Y. Chen, and L. Ding, "ETBRec: a novel recommendation algorithm combining the double influence of trust relationship and expert users," *Applied Intelligence*, vol. 52, pp. 1–13, 2021.

[2] Ya-zhi Yang, Y. Zhong, and M. Woźniak, "Improvement of adaptive learning service recommendation algorithm based on big data," *Mobile Networks and Applications*, vol. 26, pp. 1–12, 2021.

[3] X. Han, Z. Wang, and X. Hui Jun, "Time-weighted collaborative filtering algorithm based on improved mini batch K-means clustering," *Materials, Computer Engineering and Education Technology*, vol. 6258, pp. 309–317, 2021.

[4] Z. Huang, H. Ma, S. Wang, and Y. Shen, "Accurate item recommendation algorithm of ItemRank based on tag and context information," *Computer Communications*, vol. 176, 2021.

[5] T. Simpson, D. Nikolaos, and C. Eleni, "Machine learning approach to model order reduction of nonlinear systems via autoencoder and LSTM networks," *Journal of Engineering Mechanics*, vol. 147, no. 10, 2021.

[6] L. Wu, Q. Liu, W. Zhou, G. Mao, J. Huang, and H. Huang, "A semantic web-based recommendation framework of educational resources in E-learning," *Technology, Knowledge and Learning*, vol. 25, pp. 1–23, 2018.

[7] G. Sun, T. Cui, G. Beydoun et al., "Towards massive data and sparse data in adaptive micro open educational resource recommendation: a study on semantic knowledge base construction and cold start problem," *Sustainability*, vol. 9, no. 6, 898 pages, 2017.

[8] A. Morales, J. Gonzalez, K. Alquerque, J. Reyes, and S. Sanchez, "Recommendation system with graph-oriented databases for repository of open educational resources," *IOP Conference Series: Materials Science and Engineering*, vol. 1154, no. 1, 2021.
[9] B. Gan and C. Zhang, "Design of personalized recommendation system for online learning resources based on improved collaborative filtering algorithm," *E3S Web of Conferences*, vol. 214, Article ID 01051, 2020.
[10] E. Peterfreund, O. Lindenbaum, F. Dietrich et al., "Local conformal autoencoder for standardized data coordinates," *Proceedings of the National Academy of Sciences*, vol. 117, no. 49, pp. 30918–30927, 2020.
[11] L. Wang, J. Li, S. Zhang et al., "Multi-task autoencoder based classification-regression model for patient-specific VMAT QA," *Physics in Medicine and Biology*, vol. 65, no. 23, 235023 pages, 2020.
[12] W. Pan, J. Li, and X. Li, "Portfolio learning based on deep learning," *Future Internet*, vol. 12, no. 11, 202 pages, 2020.
[13] S. Chen and M. Wu, "Attention collaborative autoencoder for explicit recommender systems," *Electronics*, vol. 9, no. 10, 1716 pages, 2020.
[14] H. Sewani and R. Kashef, "An autoencoder-based deep learning classifier for efficient diagnosis of autism," *Children*, vol. 7, no. 10, 182 pages, 2020.
[15] A. B. Dincer, J. D. Janizek, and S.-I. Lee, "Adversarial deconfounding autoencoder for learning robust gene expression embeddings," *Bioinformatics*, vol. 36, no. Supplement_2, pp. i573–i582, 2020.
[16] X. Ning, Y. Qiang, Z. Zhao, J. Zhao, and J. Lian, "Tumour growth prediction of follow-up lung cancer via conditional recurrent variational autoencoder," *IET Image Processing*, vol. 14, no. 15, pp. 3975–3981, 2020.
[17] Z. Jiang, "SDAE-based feature selection method for biological Omics data," *Journal of Physics: Conference Series*, no. 1, p. 1848, 2021.
[18] N. Lu, C. Chen, W. Shi, J. Zhang, and J. Ma, "Weakly supervised change detection based on edge mapping and SDAE network in high-resolution remote sensing images," *Remote Sensing*, vol. 12, no. 23, 3907 pages, 2020.
[19] C. Li and Y. Liu, "Online dynamic security assessment of wind integrated power system using SDAE with SVM ensemble boosting learner," *International Journal of Electrical Power & Energy Systems*, vol. 125, p. 106429, 2021.
[20] Y. Liu, L. Duan, Z. Yuan, N. Wang, and J. Zhao, "An intelligent fault diagnosis method for reciprocating compressors based on LMD and SDAE," *Sensors*, vol. 19, no. 5, 1041 pages, 2019.
[21] Y. Ding, R. Dong, T. Lan et al., "Multi-modal brain tumor image segmentation based on SDAE," *International Journal of Imaging Systems and Technology*, vol. 28, no. 1, pp. 38–47, 2018.
[22] H. Yu, "Apriori algorithm optimization based on Spark platform under big data," *Microprocessors and Microsystems*, vol. 80, p. 103528, 2021.
[23] H. Mohamed, M. Tantawy Mohsen, and M. S. El Soudani Magdy, "Implementing a deep learning model for intrusion detection on Apache Spark platform," *IEEE Access*, vol. 8, Article ID 163660, 2020.
[24] K. Wang, M. M. H. Khan, N. Nguyen, and S. Gokhale, "Design and implementation of an analytical framework for interference aware job scheduling on Apache Spark platform," *Cluster Computing*, vol. 22, no. 1, pp. 2223–2237, 2019.
[25] M. Meoni, V. Kuznetsov, L. Menichetti, J. Rumševičius, T. Boccali, and D. Bonacorsi, "Exploiting Apache Spark platform for CMS computing analytics," *Journal of Physics: Conference Series*, vol. 1085, no. 3, 032055 pages, 2018.
[26] Y. Yang, J. Yu, M. Yang, P. Ren, Z. Yang, and G. Wang, "Probabilistic modeling of renewable energy source based on Spark platform with large-scale sample data," *International Transactions on Electrical Energy Systems*, vol. 29, no. 3, p. n/a, 2019.
[27] C. Fang, N. Wang, B. Yu, Y. Qin, and L. Wang, "A strategy of parallel seed-based image segmentation algorithms for handling massive image tiles over the Spark platform," *Remote Sensing*, vol. 13, no. 10, 1969 pages, 2021.