Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105
Volume **10**, Number **1**, February **2025**

Personalized Learning Resource Recommendation Based on Deep Learning and Random Forests

Dong-Hua Zheng*

School of Information Technology and Engineering
Guangzhou College of Commerce, Guangzhou 511363, P. R. China
Graduate School
Management and Science University, Shah Alam 40100, Selangor, Malaysia
20190257@gcc.edu.cn

Li-Zhu Ye

School of Information Technology and Engineering
Guangzhou College of Commerce, Guangzhou 511363, P. R. China
Graduate School
Management and Science University, Shah Alam 40100, Selangor, Malaysia
20180160@gcc.edu.cn

*Corresponding author: Dong-Hua Zheng
Received February 7, 2024, revised May 28, 2024, accepted July 28, 2024.

Abstract. *Intending to the issue that the existing single recommendation method cannot precisely express the diversity of students' interests and is not sensitive enough to the extraction of students' interest features, which results in poor outcome in recommending learning resources, a personalized learning resource recommendation method on the ground of deep learning and random forest is suggested. The method first transforms the out-of-bag data of each decision tree in the random forest in terms of the text similarity, adopts the decision tree to forecast the novel out-of-bag data, and gives the decision tree voting weights relied on the forecasting accuracy. Then the enhanced random forest algorithm is adopted to mine students' preferences for course resources, highlighting useful information and suppressing useless information through weights, further extracting key information and higher-order information of inactive users by combining the two-layer attention mechanism of Graph Attention Networks (GAN), and propagating these two kinds of information on user-course interactions to learn user and course embeddings, and, finally, combining the outcome with weighted content similarity computation fusion calculation for recommendation. Finally, the experimental outcome indicates that the designed method in this article has high NDCG, HR and Recall, and can be better applied to the field of personalized learning resource recommendation.*
Keywords: Recommended learning resources; deep learning; random forest; decision tree; graph attention network

1. **Introduction.** Personalized learning aims to enable learners to acquire appropriate learning methods and learning resources according to their learning ability, cognitive level, knowledge status, etc., which is one of the goals pursued by teaching activities for a long time [1]. Personalized learning is a customized educational method based on individual learning characteristics and needs. In recent years, with the deepening of education informatization process, all kinds of online learning platforms have been developed like mushrooming, providing learners with an open learning environment and rich learning resources, but the explosive growth in the number of learning resources, on the contrary,

leads to the learners not being able to quickly find the appropriate learning resources from the massive data, which reduces the learning efficiency [2, 3]. Therefore, how to recommend appropriate learning resources for learners according to their individual needs and preferences is an important challenge and hot issue in the field of online education [4].

1.1. **Related work.** Personalized learning resource recommendation is an important part of personalized learning, and the research directions in this field mainly include personalized recommendation algorithm, learner model construction, learning resource feature extraction and multimodal learning resource fusion. Zhu et al. [5] modeled the semantic relationship between different entities and used Euclidean distance to represent the recommendation results. Mlika and Karoui [6] proposed a recommendation model using implicit Markov chain to model the sequence of students' historical behaviors, but the recommendation results were poor. Lin et al. [7] proposed a new personalized recommendation algorithm based on support vector machine, but the running time was long. Li et al. [8] used Random Forest technology to implement a collaborative filtering recommendation method, which incorporates student feature classification and dynamic time factors. Xiao [9] used an improved collaborative filtering recommendation algorithm to recommend learning resources from the perspective of learner self-directed learning. Tawfik et al. [10] proposed a case-based recommender system to recommend similar learning resources and courses to the learners. Shu et al. [11] applied a content-based recommendation model to a programming course, but the model representation capability was insufficient.

To address the issue of low accuracy of traditional recommendation algorithms, the application of deep neural networks to learning resource recommendation has become a hot topic in current research. Bach et al. [12] used cascading null convolution to describe students' interests, but the recommendation performance was poor. Zhou et al. [13] used a variant of RNN, Long short-term memory (LSTM), to predict users' ratings of courses. LSTM is used to predict user ratings of courses. Due to the large resources consumed by LSTM, graph neural network (GNN), as a complex data structure, describes the relationship between students and learning resources in a more practical way [14]. Zhang et al. [15] introduced GNN into learning resource recommendation, and obtained a sequence vector representation combining long and short-term preferences by fusing local and global vectors, thus improving the recommendation performance. Tao et al. [16] proposed a graph attention neural network (GAN)-based item similarity model, which uses the attention mechanism to distinguish the weights of different historical items, but suffers from the cold-start problem.

Subsequently, to deal with the issues of data sparsity and cold start, hybrid recommendation algorithms have been proposed. Hao et al. [17] proposed a hybrid recommendation model based on graph attention network and SVM using multiplexed recall strategy to recommend learning resources, but the computational resource consumption is high. Balasamy and Athiyappagounder [18] proposed a multi-layer perceptron and recursive based neural network-based learning resource recommendation system to predict students' interest in course purchase, but the recommendation effect is not good. Aher and Lobo [19] used RNN and attention mechanism to extract spatio-temporal features to address the issues such as data sparsity in recommendation algorithms. Meanwhile, Hammou et al. [20] supplemented the algorithm and proposed a random forest-based learner purchase preference. The experimental outcome implies that the random forest approach has advantages over other techniques, but the recommendation results do not match their preferences.

1.2. **Contribution.** Currently, the application of recommendation to online learning resources in the field of education is still in infancy, and the use of traditional recommendation methods mainly lies in mining the static correlation between the user and the learning resources, but ignores the dynamic changes in the user's interest, leading to poor recommendation results. To deal with the above problems, this paper proposes a personalized learning resource recommendation method based on deep learning and random forest.

(1) Firstly, for the problem that the traditional random forest algorithm cannot distinguish the strong or weak predictive ability of the decision tree, the semantic information of the text vectors is used to transform the out-of-bag data, and its prediction accuracy is used as the weight of the decision tree;

(2) Then, the improved random forest algorithm is adopted to mine the dynamic preference of the students for the course resources, and then it is combined with the two-layer attention mechanism of the GAN to inhibit the useless information, and to get the higher-order information of the inactive students. The joint content similarity calculation module is adopted for recommendation.

2. Theoretical analysis.

2.1. **Graphical attention neural network.** GAN [21] adopts the way of attention mechanism to extract the features of each node in the graph, adaptively assigns different weights to each neighboring node in the way of attention, and weighted summation updates the nodes. By means of graph attention, the representation of nodes is greatly enriched, the aggregation of knowledge is realized, and the prediction efficiency of the model is improved [22]. The aggregation method of GAN is indicated in Figure 1.

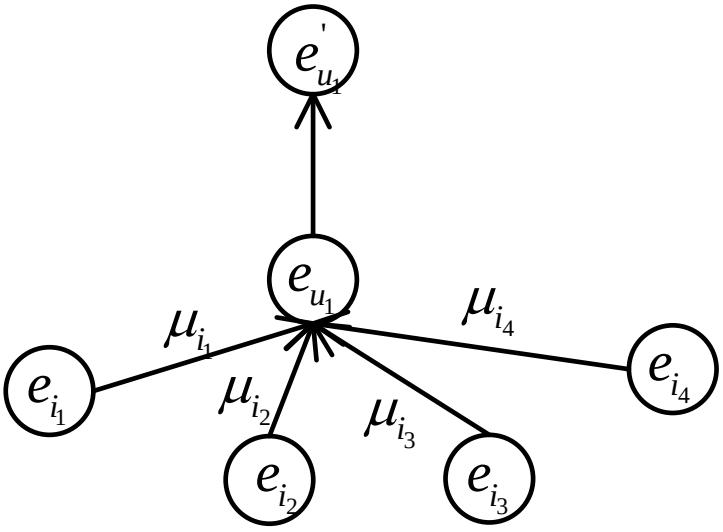

Figure 1. Polymerization of GAN

First, computing the attention factor from a central node to neighboring nodes as below.

$$\mu_{ij} = \frac{\exp(Att(h_u^{(k)}, h_j^{(k)}))}{\sum_{l \in N_u} \exp(Att(h_u^{(k)}, h_l^{(k)}))} \tag{1}$$

where μ_{ij} denotes the attention coefficient of node j to the neighboring nodes, $Att(\cdot)$ is the attention coefficient and h denotes the feature value. After obtaining the attention coefficient, neighborhood aggregation will be performed as shown in Equation (2).

$$m_u^{(k)} = \sum_{j \in N_u} \mu_{ij} h_j^{(k)} m_u^{(k)} \tag{2}$$

Neighborhood aggregation features of the target node are achieved by summing the product of the attention coefficients and the corresponding neighboring nodes, and finally, the feature output of the final node is gained as indicated below.

$$h_u^{(k+1)} = \sigma(V^{(k)} m_u^{(k)}) \tag{3}$$

where σ denotes the nonlinear activation function, V denotes the adjacency matrix, and m denotes the amount of features.

At last, nodes are aggregated and updated by Gate Recurrent Unit (GRU) [22], aggregation and updating methods are indicated in Equation (4) and Equation (5), respectively.

$$m_v^{(k)} = \frac{1}{|M_v|} \sum_{j \in M_v} h_j^{(k)} \tag{4}$$

$$h_v^{(k+1)} = GRU(h_v^{(k)}, m_v^{(k)}) \tag{5}$$

2.2. **Random forest.** Random forest through the bagging algorithm has put back sampling, from the original training samples to extract multiple subsets of samples, and based on these subsets of samples to generate multiple decision tree models; in the process of constructing the decision tree draws on the randomized feature subspace method, in all the features of some of the features for the splitting of the decision tree, and ultimately the final classifier integrated by a number of decision trees is a random forest [23].

Random forest is an integrated classification algorithm that consists of multiple decision tree-based classifiers $\{h(x, \vartheta_l), l = 1, 2, \ldots, L\}$, where $\{\vartheta_l, l = 1, 2, \ldots, L\}$ is a sequence of independent and identically distributed random vectors, and L denotes that there are L decision trees in the random forest. Given the independent variable x, after L iterations of training, a classification sequence is obtained: $\{h_1(x), h_2(x), \ldots, h_L(x)\}$. These sequences are combined into an integrated classification algorithm. Each decision tree has one vote, and the L decision trees vote to determine the category with the most votes as the final outcome.

$$H(x) = \arg\max_Y \sum_{i=1}^{L} J(h_i(x) = Y) \tag{6}$$

where $H(x)$ denotes the classification result of the random forest, $h_i(x)$ denotes the classification result of each decision tree base classifier, Y denotes the actual category, and $J(\cdot)$ denotes the schematic function.

The subset of training samples obtained by Bootstrap [24] for each decision tree in a random forest is about 63% of the training samples compared to the original training samples, while the remaining 37% of the samples are never extracted, and this part of the unextracted data is the out-of-bag (OOB) data.

3. **Optimization of Random Forest Algorithm.** In the traditional Random Forest algorithm, since the subset of feature attributes in the Random Forest is randomly chosen as the candidate subset of feature attributes for the decision tree, the resulting decision trees have different decision-making ability, which results in the weakly categorized decision trees and the strongly categorized decision trees being treated the same. In order to utilize these decision trees differently, this paper combines the semantic information of text vectors to transform the out-of-bag data, and uses the prediction accuracy of the new out-of-bag data as the weight of the decision tree. The specific steps are as follows.

(1) Assume that there are m test samples $Test_D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ and n training samples $Train_D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ in the text data, and input all the training sample sets.

(2) Generate a different training set for each decision tree. Adopting the Bootstrap sampling method, assuming that when generating the i-th decision tree, n samples are randomly sampled from $Train_D$ to form the training subset $Train_{D_i} = \{(x'_1, y'_1), (x'_2, y'_2), \ldots, (x'_n, y'_n)\}$ of the i-th decision tree, then the out-of-bag data of the decision tree is as follows.

$$\begin{aligned} OOB &= set(Train_D) - set(Train_{D_i}) \\ &= \{(oob_{x_1}, oob_{y_1}), (oob_{x_2}, oob_{y_2}), \ldots, (oob_{x_{n'}}, oob_{y_{n'}})\} \end{aligned} \tag{7}$$

where $set(\cdot)$ denotes the set of training samples.

(3) Calculate the text similarity between the test set and the out-of-bag data based on Word2vec spatial vectors, and obtain the text distance matrix $Dis = [dis_{ij}]_{m \times n}$.

(4) Transforming decision tree for out-of-bag data. Calculate the text similarity $dis_{ij} = \cos(x_i, oob_{x_j})$ based on Word2vec cosine similarity [25]. Choosing the most similar samples to the test set from the original out-of-bag data as the new out-of-bag data.

$$OOB_{new} = \{(oob_{x'_1}, oob_{y'_1}), (oob_{x'_2}, oob_{y'_2}), \ldots, (oob_{x'_m}, oob_{y'_m})\} \tag{8}$$

where $oob_{x'_r}$ is the training sample for $\min\{dis_{11}, dis_{12}, \ldots, dis_{1n}\}$.

(5) Record that the amount of feature attributes of the training set generated in Step (2) is M, and randomly select l feature as a random feature subspace of the training set, and select the optimal feature attribute from the feature subset as a split node according to the information gain evaluation criterion. The decision tree node splits until all the training samples of the node belong to the same category, then it stops.

(6) Predict the new out-of-bag data using the decision tree generated in Step (5) with the following voting weights for the i-th tree.

$$b_i = \frac{\sum_{j=1}^{m} J\{Tree_{ij} = T_j\}}{m}, \quad i = 1, 2, \ldots, L; \quad j = 1, 2, \ldots, m \tag{9}$$

where $Tree_{ij}$ denotes the prediction result of the i-th decision tree for the j-th sample in the new out-of-bag data, and T_j denotes the correct classification result for the j-th sample in the new out-of-bag data.

(7) Repeat the above steps to generate L decision trees.

(8) The classification outcome of all decision trees are assembled and weighted voting is performed. The final prediction results for the j-th test sample are as follows.

$$H(x) = \arg\max_{Y} \sum_{i=1}^{L} b_i J(h(x, \}_j) = y), \quad i = 1, 2, \ldots, L \tag{10}$$

where $h(x, \}_i)$ is the i-th decision tree model. In the summary of the weighted voting classification results of all decision trees on the test samples, the category with the largest amount of votes is adopted as the final prediction.

4. Personalized learning resource recommendation based on deep learning and random forests.

4.1. Information mining and feature extraction based on optimized random forest algorithm.

Focusing on the issue that the current single recommendation method is insufficient to extract students' interest features, which results in the poor effect of recommending learning resources, this paper proposes a lightweight personalized learning resource recommendation method on the ground of deep learning and random forest. The enhanced random forest algorithm is first adopted to mine students' preferences for course resources, then combined with the two-layer attention mechanism of graph attention network to suppress useless information and obtain the higher-order information of inactive students, and then combined with the content similarity calculation module to make recommendations. The whole framework of the model is indicated in Figure 2.

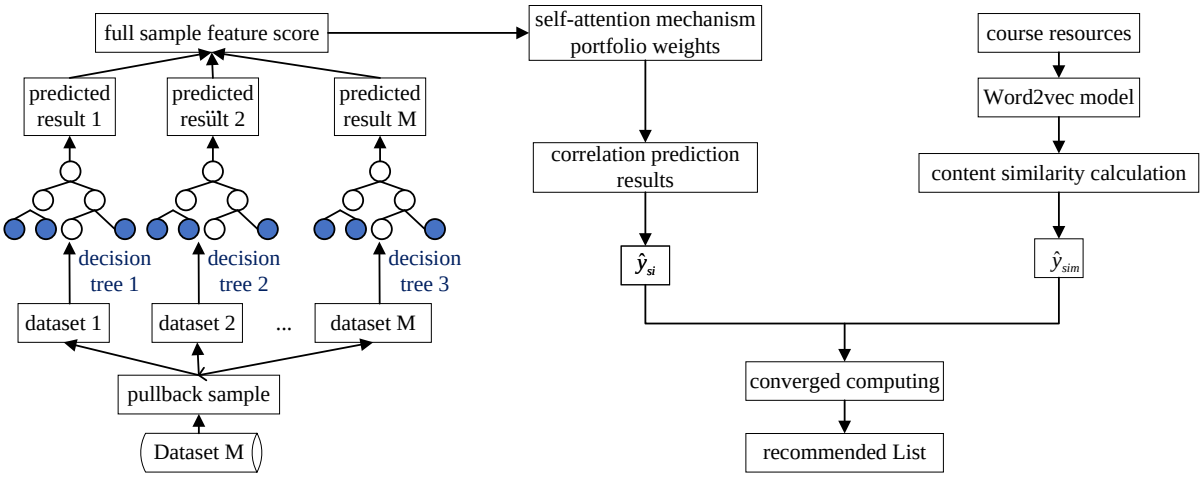

Figure 2. The whole framework of the designed model

Adopting out-of-bag estimates of the generalization error in each decision tree generation process in the improved random forest, the training samples are randomly and regressively drawn, and then the samples are perturbed to obtain the importance scores of all the sample features, which are ranked according to their scores, with the highest scores being the ones that have more relevance to the students' preferences.

Assume that the random forest model is a set of spanning trees $j(X, \delta_l), l = 1, 2, \ldots, L)$, X is $\{x_i, i = 1, 2, \ldots, M\}$, L is the amount of spanning trees, δ_l is the random variable, and assume that the training set is $Train_D = \{(x_i, y_i), i = 1, 2, \ldots, M\}$, M is the capacity of the sample, x_i is the feature vector $\{x_{i1}, x_{i2}, \ldots, x_{iN}\}$, Y is the value of the flow, and N is the features' amount in data-set. Important feature extraction's steps are as below.

(1) A sample set of L is selected randomly and with put-back from the training set, denoted as $\{Tr_{D_1}, Tr_{D_2}, \ldots, Tr_{D_j}\}$ for training.

(2) For each set of training samples to construct a decision tree, then L sets of training sets to construct L decision trees, denoted as $\{t_1, t_2, \ldots, t_l\}$. Meanwhile, owing to the principle of decision tree construction in random forests, there will be L sets of remaining data sets as out-of-bag (OOB) data not yet sampled, denoted as $\{B_1, B_2, \ldots, B_j\}$.

(3) When the out-of-bag data is when B_1, the sample set Tr_{D_1} is trained and the spanning tree t_1 is generated to predict B_1. Then the out-of-bag error for B_1 is shown in Equation (11), denoted as OOB_1, where the value of $g(x_i, \delta_1)$ is the predicted value of the survivability of t_1 when the input x_i is given.

$$OOB_1 = \frac{1}{2}\sum_{i=1}^{M}[y_i - g(x_i, \delta_1)]^2, \quad (x_i, y_i) \in B_1 \tag{11}$$

(4) Perform noise perturbation based on the j-th feature of B_1 and compute the out-of-bag error again, denoted as OOB_1^*.

(5) Repeat step (3) and step (4) for the remaining data $\{B_2, \ldots, B_j\}$, yielding OOB_1 and OOB_1^*.

(6) Calculate the importance score $Score$ using Equation (12).

$$Score = \sum_{i=1}^{l} |OOB_i - OOB_i^*| \tag{12}$$

(7) In terms of the above equation, the importance scores of all features are derived by continuously repeating Step (3) to Step (5), and the final extracted preference information features are represented as follows.

$$F = \arg\max_{m=1}^{M} \sum score_j G(f_x, Number_m) \tag{13}$$

where F denotes the student preference information features mined and extracted by the random forest algorithm; G denotes the traversal function; f_x denotes the employment information features; j denotes the voting result of the decision tree; and $Number_m$ denotes the number of features. The mining of preference information and feature extraction in the above way provide conditions for subsequent personalized recommendation.

4.2. **Personalized learning resource recommendation model based on deep learning and random forests.** The feature matrices output from each decision tree in the random forest are weighted and summed to obtain the final feature matrices E_s and E_i for the student and curriculum resource items, where the embedding weight of each layer is considered as a hyperparameter, and the inactive users need more information from higher order than the active users. Using the self-attention mechanism of graph attention network, weights are combined for different students and courses to obtain $E_s = \sum_{l=0}^{L} \alpha_s^{(l)} E_s^{(l)}$, $E_i = \sum_{l=0}^{L} \alpha_i^{(l)} E_i^{(l)}$, where $\alpha_s^{(l)}$ represents the weights of the feature vectors of the l-th decision tree in the random forest, as defined in Equation (14).

$$\alpha_s^{(l)} = soft\max(relu(g^T[E^{(0)} \parallel E^{(l)}]) \tag{14}$$

where g^T denotes the transpose of the shared attention coefficients and $relu(\cdot)$ is the activation function that limits the weight values to the range of $[0, 1]$ to avoid the problem of vanishing gradient. At the same time, the $softmax$ function is utilized to ensure that the sum of the weights of each decision tree is 1.

The final feature matrix achieved by combining the student s and the course resource item i is ranked adopting the attention function, and the items with the top-N prediction scores are written into the recommendation list, as indicated in Equation (15).

$$\hat{y}_{si} = E_s^T E_i = Att(\vec{t}_s, \vec{t}_n, \vec{t}_i) = \vec{t}_s softmax(pow(\vec{t}_s, \vec{t}_n)) \tag{15}$$

where Att denotes the attention function, t_i denotes the interaction time of item i, t_n denotes the future interest trend interaction time, and t_s denotes the future interest trend embedding.

Then the cosine similarity formula is used to calculate the content similarity between the user's items of interest and the items of the course repository, in order to obtain

the goal of precisely filtering out the courses of interest among the many recommended courses. The calculation is indicated in Equation (16).

$$\hat{y}_{sim} = \cos(\theta) = \frac{\sum_{i=1}^{m}(X_i \times Y_i)}{\sqrt{\sum_{i=1}^{m}(X_i)^2 \times \sum_{i=1}^{m}(Y_i)^2}} \tag{16}$$

where m represents the number of course resources, X, Y represent the feature vectors of the two programs, and $\cos(\theta)$ represents the cosine similarity between the two programs.

At last, the forecasting score $\hat{y}_{si}$ obtained from the design model and the forecasting score $\hat{y}_{sim}$ achieved from the similarity calculation model are fused and calculated to obtain the final prediction score, and the learning resources with higher scores are filtered according to the prediction score for recommendation.

Since $\hat{y}_{si}$ and $\hat{y}_{sim}$ are not of the same order of magnitude, $\hat{y}_{sim}$ calculated by the content similarity model is cosine similarity, and the data range is (0,1). While $\hat{y}_{si}$ is a positive integer, to facilitate the fusion calculation, $\hat{y}_{si}$ is normalized to map the data between (0,1) as indicated below.

$$\hat{y}'_{si} = \frac{\hat{y}_{si} - \hat{y}_{si,\min}}{\hat{y}_{si,\max} - \hat{y}_{si,\min}} \tag{17}$$

where the student's prediction of the resource after the normalization process is scored as $\hat{y}'_{si}$, the minimum value in the rating is denoted as $\hat{y}_{si,\min}$, and the maximum value in the rating is denoted as $\hat{y}_{si,\max}$.

$\hat{y}'_{si}$ and $\hat{y}_{sim}$ of the achieved model are weighted and summed to gain the final prediction score $\hat{y}$ of the model as follows.

$$\hat{y} = \beta\hat{y}'_{si} + (1-\beta)\hat{y}_{sim} \tag{18}$$

where β represents the value assigned to the similarity of the two weights, the larger the value of β, the greater the impact of student behavioral characteristics on the recommended results; the smaller the value of β, the greater the impact of student content preferences on the recommended outcome.

5. **Performance testing and analysis.**

5.1. **Comparison of algorithm performance.**

To estimate the effectiveness of the recommendation algorithm proposed in this paper, the MoocData dataset [26] is used to train and test the algorithm. This dataset contains real online course information, teaching video information, learning user information, etc. From the three dimensions of students, course resource items and interaction behavior, the data can be combined in various ways to support different research needs. In this paper, we mainly use the number of users, the number of course resource items and their interaction data. After preprocessing, the number of students is 627, the number of course resources is 419, and the number of student-course resource interactions is 25,729. In the experiment, the data set is randomly divided into training, testing, and validation sets according to 6:3:1 in terms of user dimensions.

For the convenience of analysis, the comparison model in the literature [20] is denoted as AEDP, the comparison model in the literature [27] is denoted as DLFS, and the algorithm in this paper is denoted as DLARF. All the experiments are carried out under the Ubuntu19.10 operating system and Python3.6 programming environment, and the parameter settings of the experiments are indicated in Table 1.

Table 1. Experimental parameter setting table

Parameter	Embedded dimension	Batch-size	Initial learning rate	Attention factor	Epoch
Value	64	2048	0.001	0.0005	40

Referring to the existing research work on recommendation models [28], this experiment uses HR@l, NDCG@l and recall@l to evaluate the metrics, where l represents the length of the recommendation list, as indicated in Table 2.

Table 2. Comparison of experimental results for different models

Method	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20
AEDP	0.5427	0.6751	0.8624	0.3481	0.4815	0.4974
DLFS	0.4996	0.6237	0.8135	0.2984	0.4236	0.4428
DLAR	0.6154	0.8126	0.9352	0.4267	0.5429	0.5831

The suggested DLAR method achieves the better performance on HR@5, HR@10, HR@20, NDCG@5, NDCG@10, and NDCG@20 metrics, which is 13.4%, 20.37%, 8.44%, 22.58%, 12.75%, and 17.23% better than the AEDP method, and 23.18%, 30.29%, 14.96%, 42.99%, and 31.68% respectively than the DLFS method. This proves the effectiveness of the model in this paper. This is because the DLFS model only recommends the most popular course resources, and does not take into account the students' attribute characteristics and interests, which results in a muchworse recommendation effect than the other two deep learning-based recommendation models. The AEDP is relied on the session recommendation, and the students' learning behaviors in a session are continuous, so the AEDP has a significant improvement in both metrics compared to the DLFS model. Therefore, the AEDP has significant improvement in both metrics compared to the DLFS. The DLAR uses the Random Forest algorithm to extract the learner's interest features, and uses multi-dimensional embedding vectors to model the students' interests in the online education platform, which is conducive to improving the accuracy of the sequence recommendation, and thus has the best recommendation performance.

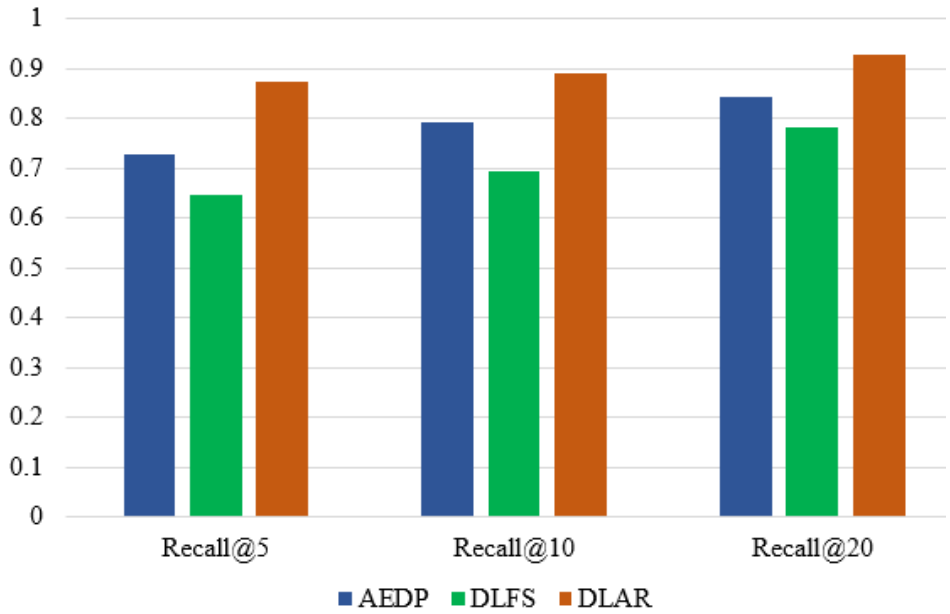

Figure 3. Recall comparison of different algorithms

A comparison of the recall of the DLAR method with the other two methods is indicated in Figure 3. When the length of the recommendation list is 5, 10, and 20, the recall of the DLAR method is 0.873, 0.891, and 0.926, respectively, while that of the AEDP method is 0.728, 0.793, and 0.843, and that of the DLFS method is 0.647, 0.692, and 0.781, respectively, which shows that the DLAR method outperforms the AEDP method and the DLFS methods. Since the graph attention neural network itself can effectively alleviate data sparsity, adding the content similarity calculation module on this basis makes up for the vacancy of a single graph neural network for content recommendation, and alleviates the problem of overly fuzzy and sparse data that most single recommendation algorithms may recommend by verifying the improvement of HR, NDCG, and Recall performance metrics.

5.2. **Comparison and analysis of the advantages and disadvantages of recommendation methods based on accuracy evaluation metrics.** In addition to visualizing the recommended performance of different methods through HR, NDCG, and Recall, the prediction effect evaluation indexes AUC, Accuracy, Precision, F1 values and Mean Square Error (MAE) of each type of method can also be further analyzed in depth to obtain more information. The prediction effect evaluation indexes of each model are summarized as indicated in Table 3.

Table 3. Comparison of the prediction effect of different recommendation methods

Method	AUC	Accuracy	Precision	F1-Score
AEDP	0.839	0.842	0.829	0.824
DLFS	0.751	0.784	0.752	0.771
DLAR	0.915	0.912	0.918	0.927

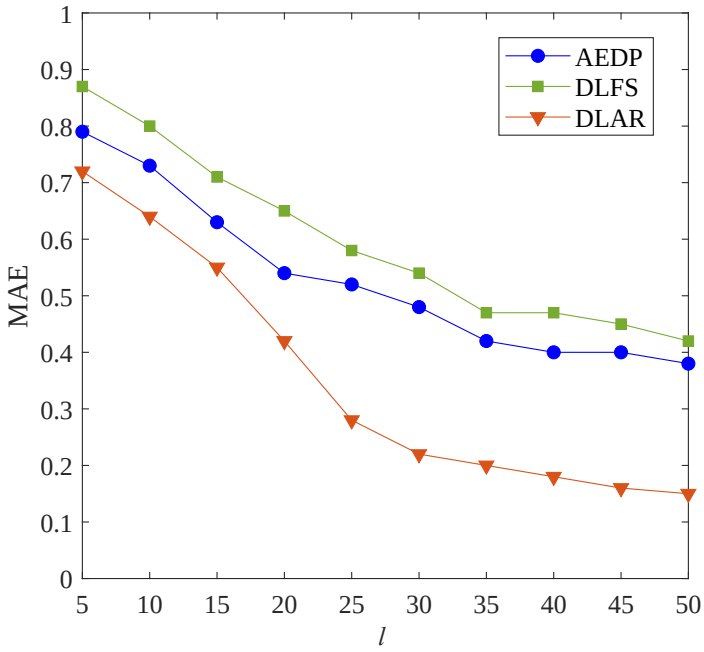

Figure 4. Comparison of MAE for different algorithms

As can be seen from Table 3, the AUC values of the three methods are all greater than 0.5, indicating that the total prediction accuracy of the three models is quite high. The

highest is DLAR with an accuracy of 0.912, followed by SCDAM with an accuracy of 0.842. These two models have an accuracy of 0.8 or more, and their prediction ability is very strong. Looking at the accuracy alone are prone to make the model prediction fall into a misunderstanding. F1 is the reconciled average of the precision rate and the recall rate, which makes the evaluation more comprehensive, neither falling into the predicament of predicting too many negative samples as positive samples and thus improving the recall rate, nor falling into the misunderstanding of predicting insufficient positive samples, which can guarantee the precision rate and the recall rate at the same time, and is more accurate. From the above table, we can see that the strongest correct prediction ability is DLAR, with an F1 value of 0.927, the F1 value of the AEDP method is between 0.8 and 0.9, and the F1 value of DLFS is the lowest, at 0.770, indicating that the DLFS method has the worst prediction performance among these three models.

As indicated in Figure 4, the MAE values of the three models gradually stabilize with the increase of the recommendation list length l, and the DLAR recommendation model always has a smaller MAE value. At the same time, comparing AEDP and DLFS, it can be seen that when l is small, the gap between the two MAE values is not large, but as l increases, the DLAR in this paper decreases faster than AEDP and DLFS, which indicates that the improved random forest algorithm extracts features and uses graph attention network to suppress useless information, making the prediction error smaller, proving that the DLAR has better performance. DLAR has better performance.

6. **Conclusion.** Focusing on the issue that existing recommendation methods only mine the static correlation between users and learning resources, but ignore the dynamic change of user interests, leading to poor recommendation effect, this article suggests a personalized learning resource recommendation method relied on deep learning and random forest. First of all, according to the text similarity transformation of the random forest of each decision tree outside the bag of data, the use of decision trees on the new bag of data prediction, and based on the prediction accuracy of the decision tree voting weight. Then the improved random forest algorithm is used to mine students' dynamic preferences for course resources, and then combined with the two-layer attention mechanism of GAN to suppress useless information, obtain the higher-order information of inactive students, and then combine with the content similarity calculation module to make recommendations. Finally, this article conducts comparative experiments on the performance of the algorithms on the MOOCData, and the experimental outcome indicates that the recommendation algorithm offered in this article has higher NDCG, HR, and recall, which verifies the effectiveness of the designed algorithm.

Acknowledgment. This work is partially supported by the 2023 Guangdong Provincial Education Research Higher Education Project (No. 2023GXJK407), 2022 Guangdong Province Education Department Project (No. 2022SJJXGG992), and 2022 School-level Quality Engineering Construction Project (No. 2022JXGG29).

REFERENCES

[1] A. Shemshack, and J. M. Spector, "A systematic literature review of personalized learning terms," *Smart Learning Environments*, vol. 7, no. 1, pp. 1-20, 2020.

[2] M. L. Bernacki, M. J. Greene, and N. G. Lobczowski, "A systematic review of research on personalized learning: Personalized by whom, to what, how, and for what purpose (s)?," *Educational Psychology Review*, vol. 33, no. 4, pp. 1675-1715, 2021.

[3] C. Walkington, and M. L. Bernacki, "Appraising research on personalized learning: Definitions, theoretical alignment, advancements, and future directions," *Taylor & Francis*, vol. 52, no. 3, pp. 235-252, 2020.

[4] N. Manouselis, R. Vuorikari, and F. Van Assche, "Collaborative recommendation of e-learning resources: an experimental investigation," *Journal of Computer Assisted Learning*, vol. 26, no. 4, pp. 227-242, 2010.
[5] X. Zhu, Y. Li, J. Wang, T. Zheng, and J. Fu, "Automatic recommendation of a distance measure for clustering algorithms," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 1, pp. 1-22, 2020.
[6] F. Mlika, and W. Karoui, "Proposed model to intelligent recommendation system based on Markov chains and grouping of genres," *Procedia Computer Science*, vol. 176, pp. 868-877, 2020.
[7] L. Lin, "Learning information recommendation based on text vector model and support vector machine," *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 2, pp. 2445-2455, 2021.
[8] X. Li, Z. Wang, L. Wang, R. Hu, and Q. Zhu, "A multi-dimensional context-aware recommendation approach based on improved random forest algorithm," *IEEE Access*, vol. 6, pp. 45071-45085, 2018.
[9] L. J. Xiao, "An improved clustering-based collaborative filtering recommendation algorithm," *Cluster Computing*, vol. 20, pp. 1281-1288, 2017.
[10] A. A. Tawfik, K. Kim, and D. Kim, "Effects of case library recommendation system on problem solving and knowledge structure development," *Educational Technology Research and Development*, vol. 68, pp. 1329-1353, 2020.
[11] 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.
[12] N. X. Bach, D. H. Long, and T. M. Phuong, "Recurrent convolutional networks for session-based recommendations," *Neurocomputing*, vol. 411, pp. 247-258, 2020.
[13] Y. Zhou, C. Huang, Q. Hu, J. Zhu, and Y. Tang, "Personalized learning full-path recommendation model based on LSTM neural networks," *Information Sciences*, vol. 444, pp. 135-152, 2018.
[14] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: a survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1-37, 2022.
[15] M. Zhang, S. Wu, X. Yu, Q. Liu, and L. Wang, "Dynamic graph neural networks for sequential recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4741-4753, 2022.
[16] Z. Tao, Y. Wei, X. Wang, X. He, X. Huang, and T.-S. Chua, "MGAT: Multimodal graph attention network for recommendation," *Information Processing & Management*, vol. 57, no. 5, 102277, 2020.
[17] J. Hao, Y. Dun, G. Zhao, Y. Wu, and X. Qian, "Annular-graph attention model for personalized sequential recommendation," *IEEE Transactions on Multimedia*, vol. 24, pp. 3381-3391, 2021.
[18] P. K. Balasamy, and K. Athiyappagounder, "An Optimized Feature Selection Method for E-Learning Recommender System Using Deep Neural Network based on Multilayer Perceptron," *International Journal of Intelligent Engineering & Systems*, vol. 15, no. 5, pp. 172-178, 2022.
[19] S. B. Aher, and L. Lobo, "Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data," *Knowledge-Based Systems*, vol. 51, pp. 1-14, 2013.
[20] B. A. Hammou, A. A. Lahcen, and S. Mouline, "An effective distributed predictive model with Matrix factorization and random forest for Big Data recommendation systems," *Expert Systems with Applications*, vol. 137, pp. 253-265, 2019.
[21] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A Spectral Convolutional Neural Network Model Based on Adaptive Fick's Law for Hyperspectral Image Classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19-46, 2024.
[22] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121-2133, 2022.
[23] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, "Application of Quantum Genetic Optimization of LVQ Neural Network in Smart City Traffic Network Prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.
[24] T. Hesterberg, "Bootstrap," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 3, no. 6, pp. 497-526, 2011.
[25] K. W. Church, "Word2Vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155-162, 2017.
[26] H. Qu, and Q. Chen, "Visual analytics for MOOC data," *IEEE Computer Graphics and Applications*, vol. 35, no. 6, pp. 69-75, 2015.
[27] H. Fang, D. Zhang, Y. Shu, and G. Guo, "Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations," *ACM Transactions on Information Systems (TOIS)*, vol. 39, no. 1, pp. 1-42, 2020.

[28] Y. Li, J.-J. Xu, P.-P. Zhao, J.-H. Fang, W. Chen, and L. Zhao, "Atlrec: An attentional adversarial transfer learning network for cross-domain recommendation," *Journal of Computer Science and Technology*, vol. 35, pp. 794-808, 2020.