

Retraction

Retracted: An English Teaching Resource Recommendation System Based on Network Behavior Analysis

Scientific Programming

Received 3 October 2023; Accepted 3 October 2023; Published 4 October 2023

Copyright © 2023 Scientific Programming. 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.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] J. Zhang, "An English Teaching Resource Recommendation System Based on Network Behavior Analysis," *Scientific Programming*, vol. 2021, Article ID 6191543, 9 pages, 2021.

Research Article

An English Teaching Resource Recommendation System Based on Network Behavior Analysis

Jing Zhang 

Faculty of Linguistic and Cultural Studies, Changchun Sci-Tech University, Changchun, Jilin 130600, China

Correspondence should be addressed to Jing Zhang; 100278@cstu.edu.cn

Received 1 October 2021; Revised 27 October 2021; Accepted 19 November 2021; Published 30 November 2021

Academic Editor: Jamil Hussain

Copyright © 2021 Jing Zhang. 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.

The sharing of English teaching resources has always been a concern. In order to further improve the value of different English teaching resources, this paper proposes a resource management system based on an improved collaborative recommendation algorithm. The proposed model can predict user behavior based on deep learning models of graph neural network (GNN) and recurrent neural network (RNN). The graph neural network can capture the hidden state of local user behavior and be used as a preprocessing step. Recurrent neural networks can capture time series information. Therefore, the model is constructed by combining GNN and RNN to obtain the advantages of both. In order to prove the effectiveness of the model, we used CNGrid's real user behavior dataset in the experiment and finally compared the results with other methods. The different deep learning-based models achieved a precision of up to 88% and outperformed other traditional models. The experimental results show that this new deep learning model has good sharing value.

1. Introduction

In today's world, with the development and technological innovation, we can see the successful participation of the recommendation system in many applications, especially web applications. It is not difficult to find that the recommendation algorithm has achieved great success. We can predict that as time goes by, the recommendation algorithm will continue to be updated, and it will be applied to more and more scientific fields, which will impact the intelligence of the entire system and the user experience. Help will be more effective. The teaching resource recommendation system is a different way of teaching assistant application from the past. In different fields, especially in education and teaching, some systems have been diversified and used, and teaching assistant systems are particularly widely used. Through this assistant system, teachers and students can communicate very conveniently, and at the same time, it can be more convenient for sharing teaching resources. Through the auxiliary system, students can request teaching resources from the teacher very accurately and conveniently, and it becomes easier and more direct for students to share or

consult. It is also for this reason that the development of recommended system modules for integrated teaching resources has a very high status in the entire teaching aid system. By using the recommendation system, the utilization rate of teaching resources will be greatly improved, especially when they can make full use of the system to share educational resources. The recommendation system recommends resources of interest to students, so that they can improve their learning efficiency and interest in learning-related knowledge, which is particularly helpful for students' learning. With the large-scale advancement of the social Internet, students can learn more and more resources, and there are obviously too many problems. A huge number of resources are constantly presented to us, but users have more difficulty finding the resources they really need, which is the information overload problem just mentioned. In order to overcome this problem, academia and society have carried out repeated experiments and large-scale research, and put forward some solutions. The recommendation system is one of the solutions. The recommendation system can efficiently provide users with resources, and these resources get rid of the previous overload problem and present the resources

that users most need. As an advanced technology, the recommendation system is customized. It can be used to analyze the information needs of users, and through the establishment of user models, the corresponding resources can be accurately pushed to the past.

The recommendation system is no longer unfamiliar to users, such as Taobao, Amazon, and Jingdong. These mainstream e-commerce websites are inseparable from the recommendation system. In addition to e-commerce, other reading websites and movie websites also use the recommendation system. Of course, it is also constantly advancing the direction in other fields, including teaching aids. The system studied in this paper will realize the function of personalized resource recommendation, which is realized through the combination of clustering algorithm and collaborative filtering recommendation algorithm. At present, we can see many types of recommendation algorithms in the industry. The most commonly used ones are based on collaborative filtering algorithms, clustering algorithms, and association rules for recommendation. At present, the most wide applicable and most mature algorithm in the industry is the collaborative filtering recommendation algorithm. At the same time, it also has detailed internal classifications, mainly two categories: the first category is based on users, and the second category is based on projects [1]. Although these two algorithms are different in some aspects, no matter which one is based on these two, they belong to the same category. Then, these two recommendation methods mainly rely on sufficient known users, which is also collaborative filtering recommendation. The user model, which can well illustrate the customer's preferences, and the preferences they usually choose, is the most important basis for judging the algorithm. The algorithm helps users find a set of users that matches the target user and has similar preferences to the target user, that is, the nearest neighbor. After analyzing the statistical data of the nearest neighbor algorithm, the data obtained by the user is obtained through weighted calculation. Then, the application engine uses the data comparison to recommend the most likely similar items to the user, that is, the student. However, there will be a problem here: the cold start problem. When a brand new user enters our recommendation system for the first time, because there is not enough data to supply the user model, there is no rating information for the user. As a result, it is impossible to recommend the most suitable items for our new users [2]. Koren et al. [2] proposed Markov decision Processes (MDP) to propose recommended items for the sessional recommendation system. The simplest MDP can be summed up as a first-order Markov chain. Its next recommendation can be simply calculated by the probability of transfer between items. Sun et al.[3] modeled the sequential behavior between every two adjacent clicks and provided more accurate predictions for each sequence. However, the main disadvantage of models based on the Markov chain is that they independently combine the components of the past. This independence assumption is too strong, which limits the prediction accuracy. On this basis, the first-generation graph convolutional neural network simply sets the convolution kernel parameter in the frequency domain as the number of

nodes for learning according to the special points of the convolution on the graph network structure. Later, Du [4] optimized the setting of the convolution kernel parameter in the frequency domain. The computational complexity of the graph convolutional neural network is reduced by fitting the convolution kernel with Chebyshev polynomial. Li et al. [5] introduced the gated unit into graph convolutional neural network. On this basis, Wu et al. [6] applied this method to user behavior analysis and achieved good results. However, although the method based on graph convolution has strong logical reasoning ability, the network performs an iterative operation according to its adjacent local nodes during message transmission. It is only applicable to the user session sequence with a short length. This topic actually needs to model according to a single user. Due to the different operation characteristics of each user, the length of the user operation session to be processed may be different in size, which challenges the model's ability to process different input data. At the same time, there is a bottleneck problem in the recommendation speed. The collaborative filtering algorithm has a bottleneck in the recommendation speed, especially when searching for the nearest neighbors. In the process, it takes a long time.

In order to solve the above problems, this paper proposes a user behavior analysis method based on a graph neural network, which applies it to the recommendation system of English teaching resources. Firstly, the graph neural network layer is used to learn the relationship between an operation and all other operations around it and generate a latent vector for each operation type, which can help learn the user's local preferences. Secondly, the recurrent neural network is used to learn the hidden information between the operation sequences. The overall workflow of the method can be described as follows: firstly, we can construct a graph data structure according to the user's operation sequence [7]. Secondly, the graph neural network is used to process the data structure to obtain the latent vectors of all operation categories. Thirdly, these operation categories are rearranged according to the input operation order, and the recurrent neural network is used to learn the arranged latent vectors to obtain the user's global operation preferences [6]. Finally, the output vector of the recurrent neural network is passed to the Softmax layer to obtain the probability of each operation type of the user in the next step. In practice, many experiments have been carried out using real user behavior operation datasets in the environment, and the experimental results have proved the effectiveness of the method.

2. System Use Case Analysis

The intelligent recommendation English teaching resource system constructed in this paper is mainly for the majority of English learners [8]. Through the resource recommendation system, English learners can not only complete the learning of English resources but also complete the download of resources; the system recommends different English learning resources to users according to the attributes of users, and at the same time manages the system, including basic user information, user permissions, and logs; and the

resource uploader uploads relevant English teaching resources through the system, including English courseware, English videos, images, and English songs. Therefore, based on the above-mentioned different roles, the overall use case diagram is shown in Figure 1.

2.1. System Function Module Design. Based on the above analysis of use cases, the functions of this system are designed into four modules: personal space management, resource management, system management, and recommendation modules. The details are shown in Figure 2.

2.1.1. Personal Space Management. The personal space management module includes space content management and space resource management. This module mainly provides an independent learning space for different English learners to meet the needs of different English learners for different resources.

2.1.2. Resource Management. Resource management is mainly for English resource uploaders (English teachers), including resource uploading and resource deletion. Teachers can upload relevant teaching resources to the system and share them with other users. At the same time, teachers can also delete part of the outdated teaching resources to achieve the purpose of resource renewal.

2.1.3. English Resource Recommendation. The recommendation module is mainly combined with the user's attributes and characteristics, combined with the user's browsing score, and then completes the purpose of recommending to the user. In this module, different recommendation models are used to recommend teaching resources for new users and old users.

2.1.4. System Management. System management is used for system user management and data maintenance. The user of this module is the system administrator. In addition to the above permissions, the system administrator also needs to divide the permissions of different roles and manage the system logs.

2.2. Overall System Architecture Design. The overall architecture is built by B/S structural mode, and the whole system architecture is divided into three layers, the application layer, the logic layer, and the data layer, as shown in Figure 3.

2.2.1. Application Layer. The application layer mainly provides the interactive interface for the user and receives the user operation request. In the application layer of this system, it includes the personalized recommendation and nonpersonalized recommendation, which mainly takes into account the settings of new registered users and old registered users in the functional interface. At the same time, through the installation of a browser in the client, users only

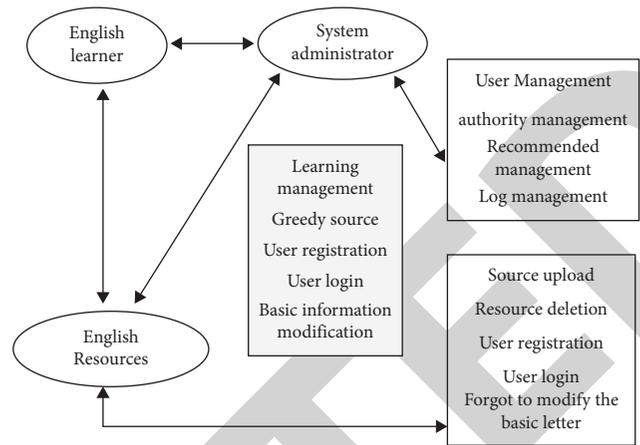


FIGURE 1: System use case analysis.

need to click the functional interface on the page to log in to the collaborative recommendation teaching resource management system and complete the access to different functions [9].

2.2.2. Logic. The logical layer mainly includes the application server and personalized recommendation model. The application server primarily responds to user action requests. After receiving the client's user request, the database server responds to the request, sends the response result back to the client, and displays it in front of the user [10].

2.2.3. Data Layer. This layer mainly contains the database management system and the database server. The database management system is used to store system data. The database server is used to respond to the application server's request and return the query results to the application server [11].

3. GR-NN User Behavior Analysis Method

3.1. Symbol Definition. As can be seen from the introduction in the previous section, the goal of user behavior analysis is to predict the next type of user action according to the known sequence of historical action types in a user session, so the basic type of user action needs to be defined first. Let the set of user operation types be $V = \{v_1, v_2, \dots, v_M\}$; that is, each operation of the user can only be a unique element in the set V . When actually organizing user session data, all operations of a session can be obtained according to the session ID segment of the data, and then all operation data of a session can be sorted by the timestamp segment of the data. Here, set the list of data in a sorted session as $S = [VS, 1, VS, 2, \dots, VS, N]$, where VS, I represents the type of operation for the user at the i -th time in the session operation list S . The goal of the user behavior anomaly detection model is to predict the type of the user's next operation. For example, according to the operation list $[VS, 1, VS, 2, \dots, VS, N-1]$, the next operation type VS, N is deduced, and then the predicted result is compared with the actual operation of the

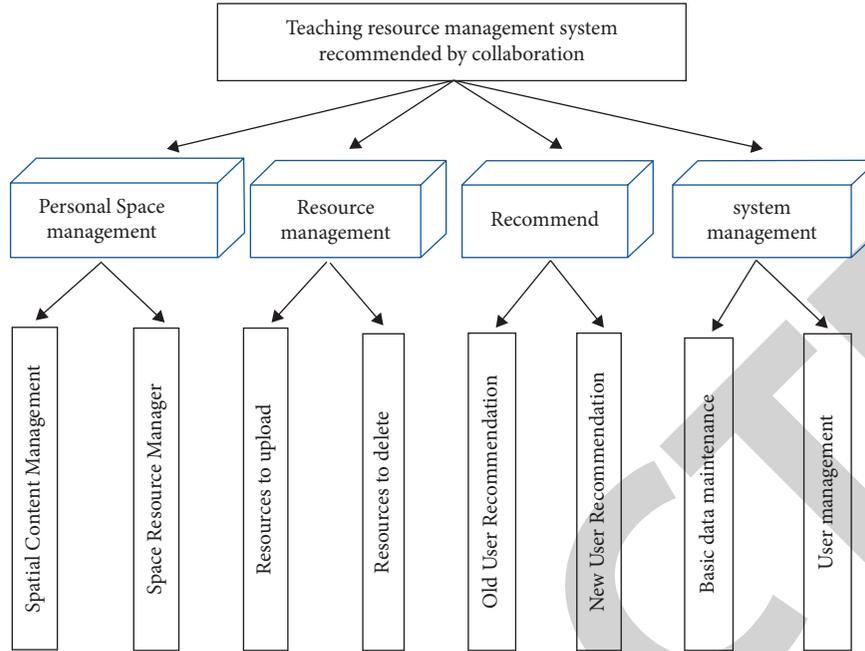


FIGURE 2: System functional modules.

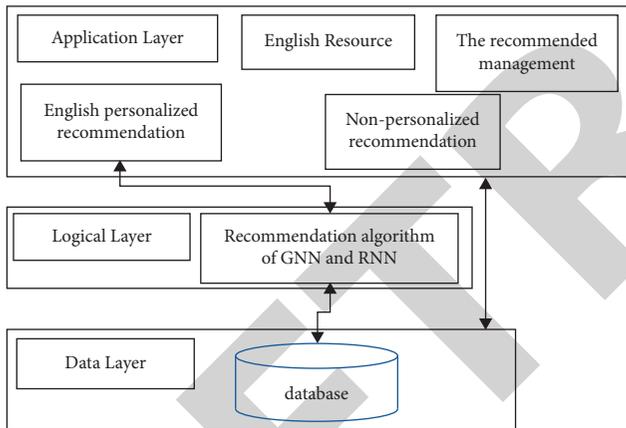


FIGURE 3: Overall architecture of the system.

user. If different, it indicates that the user may have conducted an abnormal operation.

In order to make the user sequence data suitable for the graph convolutional neural network model, the user session data needs to be transformed into the graph data structure before the input model is trained. In fact, every session can be regarded as a directed graph $G S (VS, \varepsilon S)$, in which each node represents an operation type $VS, I \in V$, and each edge $(VS, i-1, VS, I) \in \varepsilon S$ represents the user in the session after performing the $VS, i-1$ operation. Since there is a time interval between each operation, each edge can be added with an interval-based weight. Then, each entry of each session graph $v \in V$ can be embedded into a uniform dimensional Euclidian space through an embedding layer, and the node vector $V' \in R^d$ obtained through the training of the graph convolutional neural network represents the learned

potential vector, where D represents the dimension of the potential vector.

3.2. Superposition Model of Graph Convolutional Network and Cyclic Network. This section first describes in detail how to convert a user session data into inputs suitable for the graph neural network and then explains how to obtain the latent vector of user behavior operations based on the graph convolutional neural network. Suppose a user operation sequence is $[v1, v3, v5, v3, v4]$, where $\{v1: \text{login}, v2: \text{run command}, v3: \text{submit command}, v4: \text{run job}, v5: \text{view resource}\}$ represents the code of user operation type in the national high-performance computing environment. Then, this sequence represents a simple operation process of the user. Since each type of operation can only occur once in the graph structure, there are only four nodes $\{v1, v3, v4, v5\}$ when the session is converted into the graph structure. For each node to be numerically represented, each node needs to be encoded as a unique natural number. For example, $\{v1 \rightarrow 0, v3 \rightarrow 1, v4 \rightarrow 3, v5 \rightarrow 2\}$ is the code in this example. Follow this rule to transform the sequence of user actions in this example into the graph structure of the session, as shown in the leftmost structure in Figure 3. Subsequently, the session graph can be converted into a matrix representation. However, usually, the connections between the nodes of the graph are not dense and belong to a sparse graph, so COO (coordinate format) is used for storage. The matrix in the middle of Figure 4 represents the transformation of the session graph into a matrix, while the data on the right of Figure 4 shows the transformation of the graph matrix of the user session into the corresponding COO representation. Specifically, the edge_index variable uses a two-dimensional tensor to store the number of nodes

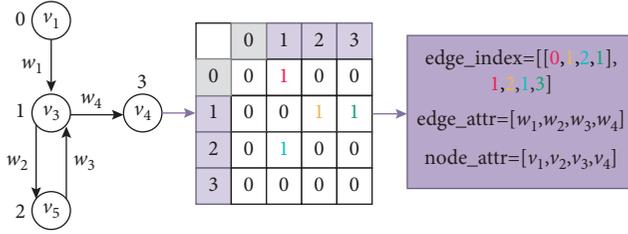


FIGURE 4: User behavior data conversion.

that appear sequentially between wired nodes, such as the first column $[0, 1]$ (highlighted in red) representing a line between node 0 and node 1 in the graph. The dimension of $edge_attr$ is the same as the number of columns in $edge_index$ and stores the weights on the corresponding lines. The dimension of $node_value$ is the same as the number of nodes, and the operation type corresponding to each node number is stored in order.

Because the time interval between the user's two operations plays a very important role in data prediction, the data is processed into the weights above the graph structure. In practice, every time data is a continuous nonnegative real number, and time data needs to be standardized. Since the range of time data is relatively large, Softmax is used here to normalize the time data. The weight calculation formula set on each edge is as follows:

$$W_i = \frac{e^{t_i}}{\sum_{j=1}^n e^{t_j}}, \quad (1)$$

where t_j represents the interval time between the two operation nodes connected by the j entry. After the input session data is transformed into the corresponding graph structure according to the above rules, a distributed representation in presentation learning is first used to construct an embedding layer, in which each category vector is transformed into a low-dimensional dense vector through the embedding layer. Then, it can be learned through the graph convolutional neural network. The key to the graph convolution operation is generalizing the convolution operator to an irregular domain, which is usually represented as a neighborhood message [12], where the passing pattern can be calculated using the following formula [13]:

$$x'_i = \gamma(x_i, \square_{j \in N(i)} \phi(x_i, x_j, e_{j,i})), \quad (2)$$

where x_i represents a differentiable, permutation invariant function. For example, sum, average, maximum, Gamma γ , and ϕ represent differentiable functions, such as multilayer perceptron commonly used in neural networks. Let the data obtained after the embedding layer be X ; then, the graph convolutional neural network proposed in [14] can be used for convolution. Its forward message passing function is as follows:

$$X' = D^{-1/2} A D^{-1/2} X \Theta, \quad (3)$$

where $A = A + I$ is inserted into the adjacent moment matrix of the self-circular ring and $D_{ii} = \sum_{j=0} A_{ij}$ is its

antiangular matrix. Θ is a trainable parameter matrix. Equation (3) actually carries out a message update for each node in the session graph. This update operation aggregates the weight data of the neighbor nodes and the corresponding edges of the node. Therefore, in this step, we learn the internal logic knowledge of the next operation corresponding to different types of operations during the session.

The graphic convolutional neural network module of the constructed model has made the latent vector learn the local transformation between each operation and the time interval information between operations through the message passing mechanism. However, the graph data structure ignores the sequential information of user operations, so to better learn the global information of users, the cyclic neural network should be further learned in the future. In fact, choosing figure convolution superposition of circulation of the neural network after network way figure is the cause of convolution layer being viewed as data preprocessing part. Compared with an ordinary convolution neural network, the figure convolution neural network can satisfy the session structure to the adjacent node number in change situation. At the same time, it can learn the logical relationship of nodes before and after the operation. However, before carrying out the cyclic neural network module, the latent vectors should first be rearranged according to the user's operation order at the time of input. Here, for simplification, symbol X is used to represent the arranged potential vectors, and the forward propagation cyclic neural network formula is as follows:

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}), \quad (4)$$

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{(t-1)} + b_{hz}), \quad (5)$$

$$n_t = \tanh(W_{in}x_t + b_{in} + r_t * (W_{hn}h_{(t-1)} + b_{hn})), \quad (6)$$

$$h_t = (1 - z_t) * n_t + z_t * h_{(t-1)}, \quad (7)$$

where h_t represents the hidden state at the t time step, $h_{(t-1)}$ represents the hidden state or initial state of this layer at the $t-1$ time step, r_t , z_t , and n_t are reset door, update door, and new door, respectively. Sigma is a sigmoid function, and the computed symbol $*$ represents the Hadamard product between matrices. Since the update gate is used to control information of the previous moment. The reset gate controls how much information is written to the current new gate NT from the previous state. Therefore, the data flows through equation (4) of the update gate and equation (5) of the reset gate to determine which information to save and discard, respectively, and to generate a new gate according to equation (6).

3.3. Prediction and Model Training. After the output of the cyclic neural network is obtained, the dimension of the vector is linearly transformed into the dimension of the operation category through a fully connected network, and the vector obtained after transformation is \hat{Z} . The output vector is

obtained by using the Softmax function of multiple classification problems to calculate

$$\hat{y} = \text{softmax}(\hat{z}), \quad (8)$$

where $\hat{z} \in R^m$ is the corresponding score for the category of the user action of all candidates and $a \in R^m$ is the probability of the next user action node appearing in session S . For each session graph, the loss function is defined as the binary cross entropy between the predicted and ground truth items. It can be obtained according to the following formula:

$$L(\hat{y}) = \sum_{i=1}^m y_i \ln \hat{y}_i + (1 - y_i) \ln (1 - \hat{y}_i), \quad (9)$$

where y represents a unique heat coding vector of the true value. Finally, the backpropagation algorithm is used to train.

3.4. Implementation and Deployment. The methods in this article are implemented using Python. In the process of model building, PyTorch [15] library is used to implement the deep learning algorithm, and PyG [13] is used to implement the graph convolutional neural network. In actual deployment, it can be divided into online and offline parts. The role of the offline part is to train and build the model according to the historical data. After the model is obtained, the model parameters will not be updated when the online detection is carried out, but the prediction will be made directly. After a period of time, the parameters of the iterative model are updated offline based on the newly collected data. When the model is predicted online, different parameter models can be called according to different users. Then, the operation of the user can be predicted and compared with the actual operation of the user. If different, the operation of the user is considered to be an abnormal behavior. The overall anomaly diagnosis process is shown in Figure 5. When the action $V1$ is input, the user behavior $V2'$ can be predicted through this model. When the actual user action $V2$ appears, the comparison is made with it. When the user's actual action $V3$ appears, it is compared with it, and so on, and the alarm threshold is set. For example, when the threshold is set to 3 errors for alarm, the alarm is carried out when $V1 \neq V1'$, $V2 \neq V2'$, and $V3 \neq V3'$ appear simultaneously.

3.5. User Behavior Data Acquisition System Based on Wireless Communication Network. According to the English teaching resources recommendation system proposed in this paper, it can get the user behavior data with the help of the wireless communication network. In this situation, we can establish a mobile English teaching resources recommendation system. The basic structure of the system is shown in Figure 6.

In Figure 6, the data acquisition and processor can receive the user behavior data from the mobile terminal, which can transmit the user behavior data. Generally speaking, there are many mobile terminals. Besides, the mobile

terminal not only is a terminal that can collect user behavior data but also has a function of mobile computing.

4. Experimental Results and Analysis and Evaluation

This chapter first demonstrates how to use this method to analyze actual user behavior data in CNGrid. Then, the proposed method is compared with other methods. Finally, the method in this paper is analyzed in detail through different experimental Settings.

4.1. The Dataset. In this paper, the actual operation content of CNGrid users is collected as the dataset, and the dataset is used to experiment and evaluate the method in this paper. The dataset covers a period of time from January to April 2019. During the preprocessing phase, all operations have been collated and encoded into unique integers. Then, a single user was selected as the primary experimental subject. The data of all other users were used as the comparative experimental subjects. It first filters out all sessions of length 1 and length greater than 50, leaving 464 single-user sessions and 910 remaining user sessions. In addition, similar to the literature, the input sequence and corresponding predictive labels are generated by segmenting the session data. Finally, the number of datasets obtained after sorting is shown in Table 1.

4.2. Baseline Algorithm. In order to evaluate the performance of this method, this method was compared with the following representative baseline methods:

- (1) GEXP3: using graph neural network and attention mechanism to predict user behavior [16].
- (2) CFIC algorithm: using recurrent neural networks and attention mechanisms to predict user behavior [17].
- (3) MLRS-CCE: the predicted action class is based on the session type to which the user's action is most similar in the previous session, where the similarity between the two sessions is calculated using the cosine similarity between the session vectors [18].
- (4) DCC: deep model for contextual collaborative learning [19].
- (5) AFPH: the training set and the first N most frequently occurring operations in the current session are selected as the algorithm for predicting the results [12].
- (6) LSTMAPI: the recurrent neural network is used directly for prediction [13].
- (7) SR-DL: the objective function of two-session sorting is optimized by stochastic gradient descent [20].

4.3. Evaluation Index and Parameter Setting. In order to compare the performance of the above methods, two evaluation indexes were selected for comparative analysis.

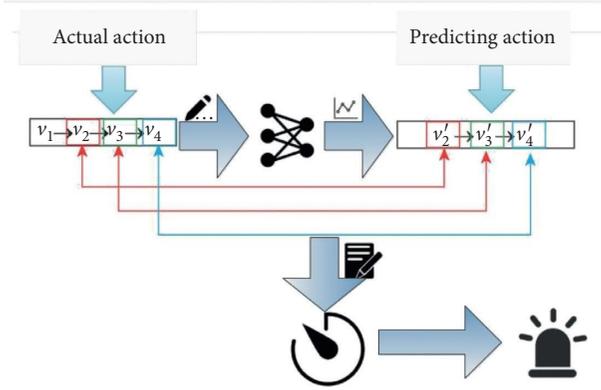


FIGURE 5: Abnormal diagnosis flowchart.

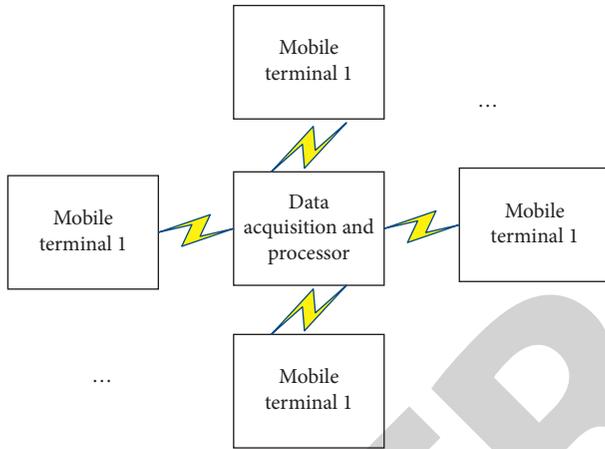


FIGURE 6: The basic structure of the system.

TABLE 1: Number of dataset.

Dataset	Train	Dev	Test
OneUser	2238	908	653
AllOthers	3250	1388	958

The metrics chosen in this article are widely used in session-based datasets [17]. Its specific description is as follows.

P@3 (precision) represents the proportion of all test data in which one of the top three prediction results with the highest probability value in the probability distribution list of model prediction results is the same as the actual result:

$$P@K = \frac{n_{hit}}{N}, \quad (10)$$

where N represents the number of all data in the test set and N_{hit} represents the number of hit data. One hit in the formula refers to the fact that the actual prediction results are included in the top K prediction results with the highest probability value in the probability distribution list of the prediction operation type.

MRR@3 (mean reciprocal rank) represents the average value of the reciprocal rank of item t within a certain range. Set to 0 if the reciprocal rank is greater than 3:

$$MRR@K = \frac{1}{N} \sum_{t \in G} \frac{1}{\text{Rank}(t)}. \quad (11)$$

MRR is a normalized score within the range $[0,1]$. The increase of this score reflects most of the “hit” prediction results, and its ranking is at the top of the probability distribution list, indicating that the corresponding prediction model has a good performance.

In order to make the experimental results fair, the parameter settings of all deep learning methods are consistent. In the data training, a small batch of ADAM optimizer was used to optimize these parameters, in which the initial learning rate was set as 0.0001, and the decay was 0.1 after every 3 iterations. The size and L2 values of data input for each training batch are set to 30 and 0.00001, respectively.

4.4. Comparison with Different Baseline Algorithms. To demonstrate the overall performance of the proposed model, the method presented in this paper was tested, and the results were calculated and then compared to the most advanced projects. The values of evaluation results of all methods are shown in Table 2, in which the best results of each column are highlighted in bold font. As shown in Table 2, on the CNGrid dataset, the performance of GR-NN in terms of P@3 and optimal level verifies the effectiveness of the proposed model.

It can also be concluded from Table 2 that the results of all the neural network baseline methods are significantly better than the traditional models, thus proving the effectiveness of deep learning techniques in this field. The SR-GNN result is slightly lower than that of the RNN method, indicating that the user’s operation sequence is usually long, and the RNN network is suitable for predicting the long user operation sequence dataset. The above deep learning methods explicitly model the user’s overall behavioral preferences and consider the transition between the user’s previous action and the next click to achieve better performance than these traditional methods. However, their performance is still lower than the method presented in this paper. The algorithm in this paper not only learns the user preferences between different operation types through the graph convolutional neural network and ignores the interference of user actions in the current session, but also can automatically select the conversion rules between the most important items through the circular neural network. In this paper, the internal logic of short sessions can be captured by GNN (graph neural network), and the long-term dependencies of long sessions can be captured by RNN. The experimental results also prove the effectiveness of this method.

4.5. Comparison with Different Graph Convolutional Neural Networks. In order to verify the correctness of the method in this paper, different GNN and RNN superpositions were selected for modeling in this round of experiments. The experimental results are shown in Table 3.

GNN methods in parentheses refer to the names of the different GNN [23] schemes selected when constructing the

TABLE 2: Experimental result of OneUser dataset.

Model	MRR@3	P@3
GEXP3	44.56	52.47
CFIC	37.9	47.37
MLRS-CCE	17.09	25.66
DCC	33.67	37.01
AFPH	58.40	81.49
LSTMAPI	68.32	88.09
SR-DL	71.23	88.40
GR-NN	67.32	85.89

TABLE 3: Result of other GNN and RNN methods.

Model	MRR@3	P@3
GR-NN(ChebConv [19])	73.57	92.99
GR-NN(GatedGraphConv [18])	73.81	93.30
GR-NN(ARMAConv [21])	73.60	93.15
GR-NN(SGConv [22])	74.64	93.46
GR-NN	74.04	93.61

overlay model. It can be seen from the above results that different GNN networks have little influence on the predicted results, which means that all GNN networks have a strong ability to learn user behavior logic and all the results in Table 3 are superior to other methods in Table 2. The result of this means for user operation of this kind of data, some actions may occur between line without a clear purpose, or in the current session user's interest in different operation between rapid drift, graph convolution neural network can effectively filter the noise in processing the session. The above discussion verifies the validity of the method in this paper.

There are several advantages of using deep learning algorithms over traditional learning and other systems and lead to better results. Deep learning algorithms are more popular than the traditional machine learning models as they can achieve more accurate results when trained using large-scale datasets. These algorithms can learn high-level features from data and may not need domain expertise and high-level features extraction. DL-based models produce the best results with unstructured data. As most of the data produced are unstructured and in different formats, we need to structure the data or develop systems that can process unstructured data. Deep learning enables us to develop decision support systems and recommendation systems that can produce better results with unstructured and unlabeled data as well. A deep learning algorithm can perform thousands of complex and repetitive tasks in a very short time when trained once and produces accurate results.

4.6. Compare the Datasets of All Other Users. In order to verify the effectiveness and advantages of the proposed method in different situations, the proposed method is applied to the datasets of all other users. The dataset of all other users contains a variety of different users whose behavior is complex and variable, so the data contains both long and short sessions, which makes the prediction more

TABLE 4: Experimental result of AllOthers dataset.

Model	MRR@3	P@3
GEXP3	43.22	50.78
CFIC	43.23	53.45
MLRS-CCE	33.74	32.78
DCC	41.87	49.16
AFPH	51.02	67.98
LSTMAPI	25.12	31.30
SR-DL	26.33	30.99
GR-NN	62.09	77.33

difficult. As shown in Table 4, GNN networks are superior to RNN networks in complex datasets because the approach models each session as a graph, allowing for transitions between different operations in the session, so it can capture more complex and implicit connection relationships between user clicks.

The experimental results using the OneUser user dataset in Table 2 show that the results of the RNN network (RNN and RNN + attention mechanism) are better than those of the GNN network (SR-GNN method), which is contrary to the results in Table 4 because the graph data structure ignores the sequential information of user operations. Therefore, only using the graph convolutional neural network is easy to ignore the complex transformation between distant user operations, while the cyclic neural network can learn this relationship. The above discussion further proves that the GNN network and RNN network have their respective adaptive ranges. The method presented in this paper achieves the best results in both cases, which also verifies the generality of the method presented in this paper.

5. Conclusion

At present, GNN and RNN have their own advantages for the sequential prediction of user behavior. The advantage of GNN is that it can capture the potential logical relationship between nodes of the data graph, and its message is transmitted iteratively according to its adjacent local nodes, so it is suitable for short-length user session sequences. The advantage of RNN lies in its long-term memory ability, which is suitable for long user session sequences. In this paper, a method combining GNN and RNN is proposed for abnormal detection and analysis of user behavior. Since the constructed model combines GNN and RNN, the advantages of both are absorbed. A large number of experiments are carried out on the actual operation of users in the CNGrid dataset, and it is proved that this method can surpass the current mainstream methods in terms of different evaluation indexes. In the experiment, different GNN methods and different datasets are tested, and the experimental results also prove the rationality and wide applicability of the proposed model. The method proposed in this paper is based on the artificial neural network. Therefore, the method proposed in this paper can make full use of the advantages of the artificial neural network, and its performance is good. However, due to the high computational complexity of the artificial neural network, the overhead of

the system is very large, and its time complexity will be very high. Therefore, it is necessary to continue to study the systems that can be applied in practice in the future.

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 that no conflicts of interest regarding this paper.

References

- [1] Y. Koren and R. M. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds., Springer, Berlin, Germany, 2011.
- [2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
- [3] Z. Sun, M. Anbarasan, and D. P. Kumar, "Design of online intelligent english teaching platform based on artificial intelligence techniques," *Computational Intelligence*, vol. 37, 2020.
- [4] H. Du, "An english network teaching method supported by artificial intelligence technology and WBIETS system," *Scientific Programming*, vol. 2021, Article ID 8783899, 9 pages, 2021.
- [5] R. Y. Li, M. Tapaswi, R. J. Liao, J. Jia, R. Urtasun, and S. Fidler, "Situation Recognition with Graph Neural networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision*, pp. 4183–4192, Venice, Italy, October 2017.
- [6] F. Wu, T. Zhang, A. H. Souza, C. Fifty, T. Yu, and Q. Kilian, "Simplifying graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 6861–6871, Long Beach, CA, USA, June 2019.
- [7] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, "Graph neural networks with convolutional ARMA filters," 2019, <https://arxiv.org/abs/1901.01343>.
- [8] Y. K. Tan, X. X. Xu, and Y. Liu, "Improved recurrent neural networks for session- based recommendations," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 17–22, Boston, MA, USA, September 2016.
- [9] Z. A. Feng, C. Debao, X. Qingzheng, J. Ziqi, and K. Jiahui, "A two-stage personalized recommendation based on multi-objective teaching-learning-based optimization with decomposition," *Neurocomputing*, vol. 452, pp. 716–727, 2021.
- [10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations*, pp. 1–13, Toulon, France, April 2017.
- [11] Y. Cui, L. Zhang, Y. Hou, and G. Tian, "Design of intelligent home pension service platform based on machine learning and wireless sensor network," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 2, pp. 2529–2540, 2021.
- [12] J. Huang, "An internet of things evaluation algorithm for quality assessment of computer-based teaching," *Mobile Information Systems*, vol. 2021, Article ID 9919399, 10 pages, 2021.
- [13] L. I. Guangjie, H. Liu, L. I. Ge, S. Shen, and H. Tang, "LSTM-based argument recommendation for non-API methods," *Science China(Information Sciences)*, vol. 63, no. 9, p. 190101, 2020.
- [14] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. Dahl, "Neural message passing for quantum chemistry," 2017, <https://arxiv.org/abs/1704.01212>.
- [15] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," 2019, <https://arxiv.org/abs/1903.02428>.
- [16] N. Gutowski, T. Amghar, O. Camp, and F. Chhel, "Gorthaur-EXP3: bandit-based selection from a portfolio of recommendation algorithms balancing the accuracy-diversity dilemma," *Information Sciences*, vol. 546, pp. 378–396, 2021.
- [17] A. Yassine and L. Mohamed, "Intelligent recommender system based on unsupervised machine learning and demographic attributes," *Simulation Modelling Practice and Theory*, vol. 107, Article ID 102198, 2020.
- [18] J. A. Saez and E. Corchado, "A meta-learning recommendation system for characterizing unsupervised problems: on using quality indices to describe data conformations," *IEEE Access*, vol. 7, pp. 63247–63263, 2019.
- [19] C. A. Lei, Z. B. Lu, C. Shanshan, W. Zhiang, and C. Jie, "Personalized itinerary recommendation: deep and collaborative learning with textual information," *Expert Systems with Applications*, vol. 144, Article ID 113070, 2020.
- [20] M. Ludewig, N. Mauro, S. Latifi, and D. Jannach, "Empirical analysis of session-based recommendation algorithms," *User Modeling and User-Adapted Interaction*, vol. 31, no. 5, pp. 1–33, 2021.
- [21] P. Adam, G. Sam, M. Francisco et al., "PyTorch: an imperative style, high- performance deep learning library," in *Proceedings of the Annual Conference on Neural Information Processing Systems*, vol. 5, pp. 8024–8035, Vancouver, Canada, December 2019.
- [22] S. Rendle, C. Freudenthaler, Z. Gantner, and T. Lars, "BPR: Bayesian personalized ranking from implicit feedback," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pp. 452–461, Montreal, Canada, June 2009.
- [23] S. M. Chowdhury and A. Hossain, "Impact of error control code on characteristic distance in wireless underground sensor networks," *IET Communications*, vol. 12, no. 13, pp. 1540–1549, 2018.