# Exercise Recommendation Algorithm Based on Reinforcement Learning

Simiao Yu, Ji Li, Tiancheng Zhang

*Abstract*—**In recent years, the rapid development of technologies such as artificial intelligence, cloud computing, data mining, and mobile internet has significantly transformed educational methodologies. Online learning, an emerging educational paradigm, has garnered considerable attention for its accessibility and convenience. Although online education platforms offer learners a high degree of freedom, they lack personalized guidance, leading to challenges such as "Information Overload" and "Knowledge Loss". These challenges primarily manifest as learners struggling to identify resources that match their preferences and needs, adversely affecting their learning efficiency and outcomes. To address these challenges, this paper proposes an exercise recommendation algorithm that combines the emerging fields of reinforcement learning and recommendation systems to help learners find suitable exercise resources. Initially, we use a knowledge tracking model to assess the potential knowledge level of learners. Following this, the Deep Q Network algorithm is utilized to modify learners' exercise records by removing unsatisfactory exercises that learners may have mistakenly selected during the learning process. Based on the modified record of exercises and the knowledge levels of learners, the algorithm recommends appropriate exercises. Finally, extensive experiments have demonstrated the effectiveness of our method.**

*Index Terms*—**online education, knowledge tracking, reinforcement learning, exercise recommendation, Deep Q Network**

## I. Introduction

With the development of emerging information and communication technologies, including mobile communication, the Internet, the Internet of Things, cloud computing, Big Data, and artificial intelligence, human ways of thinking, production, living, and learning are undergoing significant changes. Modern education is evolving towards a paradigm characterized by networking, digitalization, personalization, ubiquity, and intelligence. Numerous new educational models have emerged, such as mobile learning, generalized learning, intelligent learning, and blended learning.

Online learning has gained significant traction in recent years, standing out as a favored avenue for personalized education. It has drawn in a multitude of learners thanks to its user-friendly nature, inclusiveness, and the vast array of educational materials available. In the realm of these modern, web-based educational platforms, learners are granted greater freedom in managing their study schedules, are exposed to a multitude of learning methodologies, and have access to an extensive pool of resources. They can independently tailor their learning process to suit their individual situations and preferences.

However, online education platforms, in contrast to traditional classrooms, lack the capability to offer real-time supervision and direct guidance to learners, leading to challenges such as "Information Overload" and "Knowledge Loss". These problems primarily manifest in learners often needing to spend a considerable amount of time finding suitable learning resources amid an extensive array of resources of varying quality. Consequently, exercise recommendation algorithms, designed to provide guidance and assistance to learners, have increasingly become a significant research focus.

One direction of exercise recommendation algorithms is based on the collaborative filtering algorithm [1]. These methods take into account learners' characteristics and preferences, making recommendations based on similar users or exercises. Salehi *et al.* [2] considered the characteristics and learning order of learners and learning resources, and recommended high-quality learning resources through both implicit and explicit collaborative filtering algorithms. Segal *et al.* [3] combined the collaborative filtering algorithm with Social Choice theory and proposed an algorithm to customize learning resources and examinations for learners. Zhao *et al.* [4] improved the efficiency of recommendations by utilizing attribute information such as user gender and topic interests.

Another direction of exercise recommendation algorithms is based on knowledge tracking algorithms. These algorithms assess learners' potential knowledge levels from their exercise histories, forecast their potential performance with different learning resources, and then recommend exercises to address their specific learning deficits. Hudak *et al.* [5] recommend learning resources by analyzing learners' learning processes and current states. Dwivedi *et al.* [6] assess learners' knowledge levels based on their grades and recommend elective courses accordingly.

Nevertheless, prevailing algorithms often concentrate exclusively on either the preferences or the knowledge levels of learners. We contend that optimal exercise recommendations should consider both aspects: learners' preferences and their knowledge requirements. Furthermore, amidst a plethora of exercise options, learners may inadvertently opt for exercises that do not resonate with their genuine interests, which could result in a lack of earnest engagement with these exercises. Consequently, an

Manuscript received Aug 12, 2023; revised Aug 4, 2024.

Simiao Yu is a lecturer at the University of Science and Technology Liaoning, Anshan, China,114000 (e-mail: 1115063992@qq.com).

Ji Li is a PhD candidate at the Northeastern University, Shenyang, China,110000. (e-mail: 408567077@qq.com).

Tiancheng Zhang is an associate professor at the Northeastern University, Shenyang, China,110000. (e-mail: tczhang@mail.neu.edu.cn).

overreliance on exercise data to assess learners' preferences and knowledge levels may introduce considerable inaccuracies into the recommendation process.

In conclusion, this paper proposes an exercise recommendation algorithm based on reinforcement learning, named RLER (Reinforcement Learning Exercise Recommender). RLER utilizes reinforcement learning to filter out exercises that learners have mistakenly selected. The specific contributions of this paper are as follows:

1) We consider learners' preferences and learning levels, enhancing the precision of learner modeling.
2) We have developed a model that leverages the Deep Q Network algorithm to modify learners' exercise records. This model is designed to filter out exercises that were erroneously selected, enabling the algorithm to make more accurate exercise recommendations based on the modified records and the learners' potential knowledge levels.
3) We compared the RLER model with five advanced exercise recommendation algorithms in two real datasets. The results of these experiments robustly demonstrate the superior effectiveness of the RLER model.

## II. Related Works

### A. Recommendation Algorithm

Personalized recommendation algorithms primarily recommend satisfactory items to users based on their preferences. Common recommendation algorithms can be categorized into three types: content-based, collaborative filtering-based, and deep learning-based recommendation algorithms.

The content-based recommendation algorithm is one of the earliest used. It was initially applied to address the personalized recommendation problem in the Fab system and later extended to fields such as music recommendation systems, e-commerce recommendation systems, and news recommendation systems. The underlying concept of these algorithms is straightforward: they recommend items to users that are similar to items they have liked in the past. Content-based recommendation algorithms require only user preference and item feature information; they do not rely on user evaluations of the items. For newly added items, these algorithms can extract features to recommend new items, effectively addressing the cold-start problem for new items and avoiding issues with sparse rating information. However, these recommendation algorithms still face the cold-start problem for new users because the content-based recommendation model cannot obtain users' interest models.

The recommendation algorithm based on collaborative filtering is currently the most widely used method. This algorithm relies on user preference evaluation data for items and predicts items that the user may like. A typical collaborative filtering algorithm involves establishing a scoring matrix containing $m$ users and $n$ items, then calculating missing scores for items using existing scores in the matrix to make recommendations.

The recommendation algorithm based on deep learning extracts latent features of users and items, generating recommendations based on these features. He *et al.* [7] proposed a neural network-based collaborative filtering model. The fundamental concept of this model is similar to traditional collaborative filtering algorithms, which involve computing the similarity between users and items. This model simulates user-item interactions through a multilayer perceptron, where the output of one layer serves as the input for the next.

### B. Knowledge Tracking Model

Knowledge tracking is a widely used technology in personalized guidance. Its task is to automatically track changes in a learner's knowledge level based on their historical learning trajectory, accurately predict their performance in future learning activities, and provide corresponding assistance.

The knowledge tracking task can be formalized as follows: given a learner's historical learning interaction sequence $X_t = (x_1, x_2, \cdots, x_t)$ on a specific learning task, the objective is to predict the learner's performance on the subsequent interaction $x_{t+1}$. Each interaction $x_t$ is characterized as $(q_t, a_t)$, where $q_t$ represents the exercise chosen by the learner at time $t$, and $a_t$ represents the answering situation of the learner at time $t$. Knowledge tracing models can be roughly divided into those based on probabilistic graphical models [8], matrix factorization, and deep learning [9].

### C. Reinforcement Learning

Reinforcement learning is an important branch of machine learning. Unlike supervised and unsupervised learning, reinforcement learning autonomously learns through interaction with the environment. Due to its robust performance in managing intricate decision-making problems that require dynamic interaction and long-term strategizing, reinforcement learning has found extensive applications in fields such as robotic control [10] and game design [11].

The standard reinforcement learning model includes four basic elements: environment, action, reward and status. The interaction process between the agent and the environment can be summarized as follows: The agent chooses an action $a_t$ in the current state $S_t$.The environment calculates the state $S_{t+1}$ of the agent at the next moment according to the action selected by the agent and provides the agent with a reward value $r_t$.The agent assesses the quality of its chosen action based on the reward value and continues to select actions in the succeeding state, persisting in this process until the termination condition is met.

Traditional value-based or strategy-based reinforcement learning algorithms are limited in that each state and action is marked by a unique identifier. This limitation leads to problems such as large storage requirements, long training times, and poor training outcomes when the state space is too large. Consequently, researchers have shifted their focus to exploring the potential of neural networks to address the challenges associated with traditional reinforcement learning. The DeepMind team ingeniously integrated neural networks with the Q-Learning algorithm, proposing the DQN (Deep Q Network) algorithm [12]. This innovative approach aims to mitigate the substantial spatial and temporal demands associated with the Q-Learning algorithm.

## III. Method

### A. Symbol Definition

The symbols in this paper are provided in Table I.

TABLE I
SYMBOL DEFINITION AND MEANING

| Symbol Definition | Symbol Meaning |
|---|---|
| $S = \{S^1, S^2, \cdots, S^M\}$ | learners' historical learning records |
| $S^i = \{S_1^i, S_2^i, \cdots, S_t^i\}$ | historical learning records of learner $i$ |
| $S_t^i = \{e_t^i, a_t^i\}$ | exercises and performances chosen by learner $i$ at time $t$ |
| $e_t^i$ | exercise chosen by learner $i$ at time $t$ |
| $a_t^i$ | performance of learner $i$ at time $t$ |
| $E = \{E^1, E^2, \cdots, E^M\}$ | learners' historical exercise records （excluding learners' performance） |
| $E^i = \{E_1^i, E_2^i, \cdots, E_t^i\}$ | historical exercise records of learner $i$ |
| $K^i = \{k_1^i, k_2^i, \cdots, k_N^i\}$ | potential knowledge level of learner $i$ |
| $\bar{S}_t^i$ | one-hot vector representation of learner $i$ 's learning records at time $t$ |
| $\tilde{E}^i$ | low-dimensional vector representation of learner $i$ 's exercise records |
| $\hat{E}^i$ | exercise record of learner $i$ after modification |

Specifically, compared to exercise records, historical learning records also include learners' performance.

### B. Model Overview

This paper proposes an exercise recommendation method that utilizes the reinforcement learning DQN algorithm, referred to as RLER. The structure of the RLER model is shown in Figure 1:

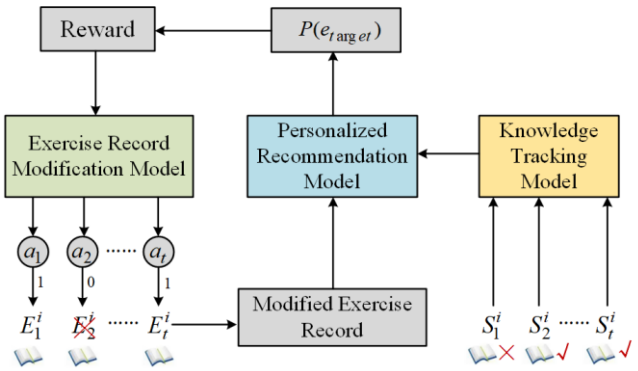


Fig. 1. Structure of RLER model

This model consists of three parts: the knowledge tracking model, the personalized recommendation model, and the exercise record modification model. The knowledge tracking model calculates the learner's potential knowledge level, which is then used to construct features for the personalized recommendation model and represent the state in the exercise record modification model. The personalized recommendation model recommends suitable exercises for learners and provides the reward function for the exercise record modification model. The exercise record modification model adjusts the learner's historical exercise records based on the reward function provided by the personalized recommendation model, determining whether the modifications are beneficial or detrimental, with the aim of enhancing the accuracy of exercise recommendations.

The flow chart of the RLER model is shown in Figure 2:

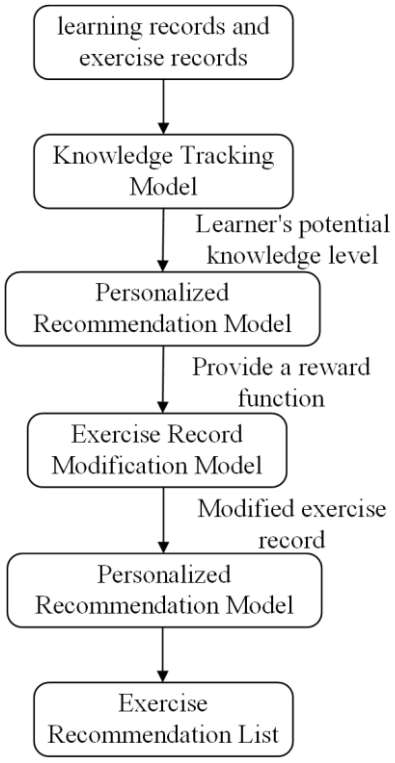


Fig. 2. Flow chart of RLER model

### C. Knowledge Tracking Model

Common personalized recommendation models, whether they are matrix factorization models, recurrent neural network models, or models utilizing attention mechanisms, all address the problem of exercise recommendation by modeling learner features based on their exercise records. However, they do not take into account the learner's performance on the exercises, which might lead to the following issues.

Assuming that learner $i$ and learner $j$ have similar exercise records, but their performance on the exercises differs. If learner $i$ completes most of the exercises correctly, and learner $j$ completes most of them incorrectly, the exercises they choose next are likely to be different.

Therefore, constructing a learner's profile based solely on the exercises they choose is not sufficiently accurate. RLER takes into account the learner's knowledge level when modeling their characteristics.

RLER employs the DKT model [9], a knowledge tracking model based on the LSTM (Long short-term memory network). The DKT model can assess the learner's potential knowledge level through their performance on learning records. The structure of the DKT model is shown in Figure 3.

Input of DKT model is the learner's learning record $S^i = \{S_1^i, S_2^i, \cdots, S_t^i\}$, the learning record of the learner $i$ at time $t$ is

specifically expressed as $S_t^i = \{e_t^i, a_t^i\}$ . $e_t^i$ represents the exercise selected by learner $i$ at time $t$, and $a_t^i$ represents the performance of learner $i$ on the exercise (1 means that the exercise is done correctly, 0 means that the exercise is done wrong). First, convert $S_t^i$ into a one-hot vector through one-hot encoding and record it as $\bar{S}_t^i$, and input it into the LSTM.

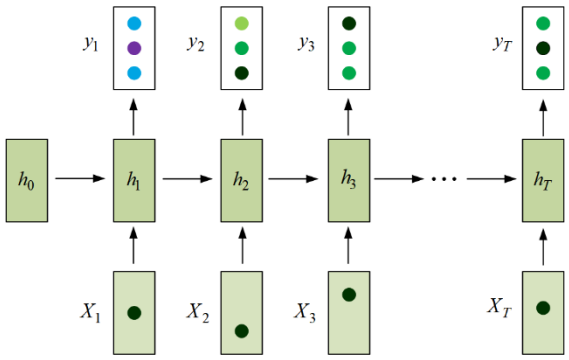


Fig. 3. DKT model

LSTM is an enhanced recurrent neural network that can address the issue of RNN's inability to handle long-distance dependencies. The structure of LSTM is shown in Figure 4.

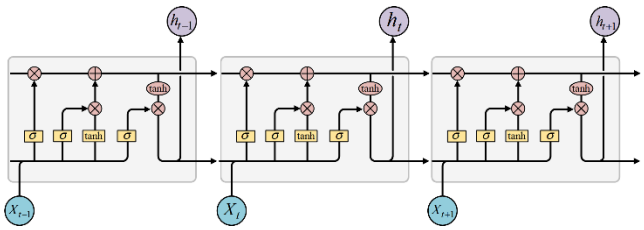

Fig. 4. Structure of LSTM

By using LSTM, the DKT model can comprehensively consider learners' past and current performance to determine their potential knowledge level. The forget gate $f_t$ in LSTM is consistent with the idea that learners tend to decrease their mastery of previously learned knowledge over time. The DKT model inputs the features extracted by LSTM into the hidden layer, and finally outputs the prediction result from the output layer.

The output of DKT represents the probability of learners answering each exercise correctly, which is recorded as the potential knowledge level of learner $i$ denoted by $K^i = \{k_1^i, k_2^i, \cdots, k_N^i\}$.

### *D. Personalized Recommendation Model*

The personalized recommendation model constructed in this paper has two functions: recommending exercises for learners and providing the reward function for exercise record modification model. The recommend-dation model mainly consists of three parts: Embedding layer, GRU layer, and Fully Connected layer. The structure of the personalized recommendation model is shown in Figure 5.

The function of the Embedding layer is to map the record of exercises completed by learner $i$ denoted as $E^i = \{E_1^i, E_2^i, \cdots, E_t^i\}$ into a low-dimensional vector $\tilde{E}^i = \{\tilde{E}_1^i, \tilde{E}_2^i, \cdots, \tilde{E}_t^i\}$.

The GRU layer is a gated recurrent unit layer, which is an improved recurrent neural network model. Its function is to extract the sequence features of exercise records. GRU has two operations: update gate and reset gate. The GRU layer calculates the output of the reset and update gates based on the input at the current time and the network hidden state at the previous time. It then computes the candidate hidden state according to the input at the current moment and the output of the reset gate. Finally, the final hidden state is obtained based on the candidate hidden state and the updated gate output, and the current output is derived from the hidden state.

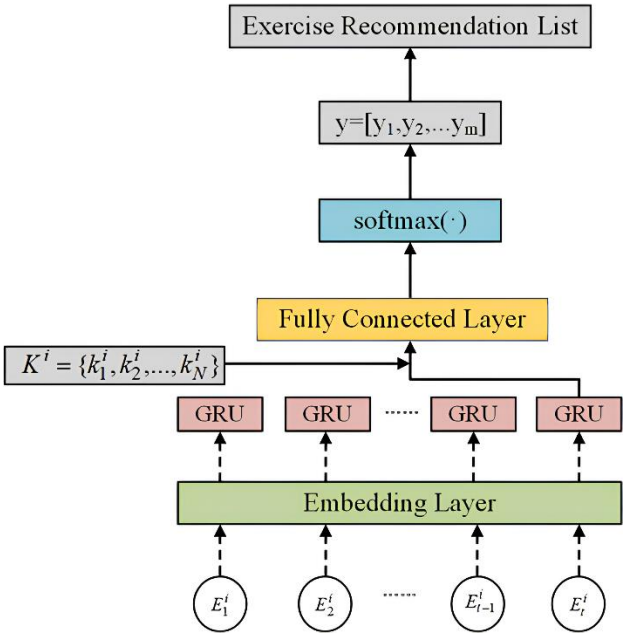


Fig. 5. Structure of personalized recommendation model

The update gate in the GRU determines the extent to which the state information from the previous time step and the current input is retained and passed to future time steps. The calculation formula is given in (1).

$$z_t = \sigma\left(W_z \cdot \left[h_{t-1}, \tilde{E}_t^i\right]\right) \tag{1}$$

Where $\tilde{E}_t^i$ represents the low-dimensional vector representation of the exercises completed by learner $i$ at time $t$. $h_{t-1}$ represents the hidden state information at time $t-1$. $W_z$ represents the weight of the update gate. $\sigma(\cdot)$ is activation function.

The reset gate of the GRU determines the amount of state information to be forgotten at the previous moment. The calculation formula is shown in (2).

$$r_t = \sigma\left(W_r \cdot \left[h_{t-1}, \hat{E}_t^i\right]\right) \tag{2}$$

Where $W_r$ represents the weight of the reset gate.

The calculation formula of the current memory content is shown in (3).

$$\tilde{h}_t = tanh\left(W_h \cdot \left[r_t * h_{t-1}, \hat{E}_t^i\right]\right) \tag{3}$$

Where $W_h$ represents the weight of the hidden layers. The product of the corresponding elements of the output of the reset gate $r_t$and the hidden state information $h_{t\text{-}1}$ determines the information to be retained at the previous moment. * represents the matrix dot product.

The calculation formula of the final memory of the current time step is shown in (4).

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{4}$$

Where $(1-z_t)*h_{t-1}$ represents the amount of information from the previous moment retained to the final memory at the current moment, and $z_t*\tilde{h}_t$ represents the amount of the current memory content retained to the final memory at the current moment.

The role of the Fully Connected layer is to calculate the probability of each exercise being selected based on the characteristics of learner $i$. The calculation formula is shown in (5).

$$y = softmax\left(W_j \cdot [K^i, h_t] + b_j\right) \tag{5}$$

Where $W_j$ is the weight of the fully connected layer.$b_j$ is the bias of the fully connected layer. $K^i = \{k_1^i, k_2^i, \cdots, k_N^i\}$ is the potential knowledge level of learner $i$ calculated by the DKT model. $[K^i, h_t]$ is to splice the potential knowledge level with the learner characteristics output by the GRU layer. $softmax(\cdot)$ is the activation function, which limits the output value between 0-1.

The personalized recommendation model employs cross-entropy as the loss function for training. The calculation is detailed in (6).

$$L(p,q) = -\sum_{i=1}^{M} p_i \, log(q_i) \tag{6}$$

Where $M$ is the number of learners. $p_i$ is the real probability of learner $i$'s choice of exercises at the next moment. $q_i$ is given by the recommendation model to represent the predicted distribution of learner $i$'s choice of exercises at the next moment.

### *E. Exercise Record Modification Model*

The exercise record modification model modifies learners' exercise records by removing as many unsatisfactory exercises as possible that learners may have mistakenly selected.

Reinforcement learning generally consists of four parts, namely Action, State, Reward, and Algorithm. The details of the introduction will be explained next.

1) Action

The exercise record modification model is to eliminate the exercises that the learners are not satisfied with. Thus, the action $a_t$ of each step has only two values. $a_t = 0$ means to delete the exercise in the exercise record. $a_t = 1$ means to keep the exercise.

2) State

The state of learners is represented by (7).

$$S = [k_1, k_2, \ldots, k_N, p_1, p_2, \ldots, p_N] \tag{7}$$

Where $k_1, k_2, \ldots, k_N$ represents the potential knowledge level of the learner, provided by the knowledge tracking model. $p_1, p_2, \ldots, p_N$ is the low dimensional vector representation of the learner's exercise record and position identifier, which is used to record the modified position.

3) Reward

The reward function of the exercise record modification model is given by the personalized recommendation model. The reward function is shown in (8).

$$Reward = \begin{cases} p\left(e_{target}\middle|\hat{E}^i\right) - p\left(e_{target}\middle|E^i\right) & if\ complete \\ 0 & otherwise \end{cases} \tag{8}$$

Where $e_{target}$ is the exercise that the learner actually chooses at the next moment. $p\left(e_{target}\middle|E^i\right)$ represents the probability of selecting the target exercise according to the modified exercise record. $p\left(e_{target}\middle|E^i\right)$ represents the probability of selecting the target exercise based on the original exercise record.

The exercise record modification model adopts the Monte-Carlo update strategy. The reward function is obtained only after the modification of the entire learning record of a learner is completed, at other times, the reward function is set to 0.

4) Algorithm

The exercise record modification model adopts the Deep Q Network algorithm, which combines a neural network with the Q-Learning algorithm. The structure of DQN is shown in Figure 6.

The loss function for training and updating the parameters of the DQN model is based on the squared difference between the actual value and the predicted value. This loss function is represented in (9).

$$L(\theta) = \left(r_t + \gamma \, max_a \, Q_{\bar{\theta}}\left(s_{t+1,}a\right) - Q_\theta(s_t, a_t)\right)^2 \tag{9}$$

Where $Q_\theta(s_t, a_t)$ represents the predicted value of the reward that will be obtained by choosing action $a_t$ in the

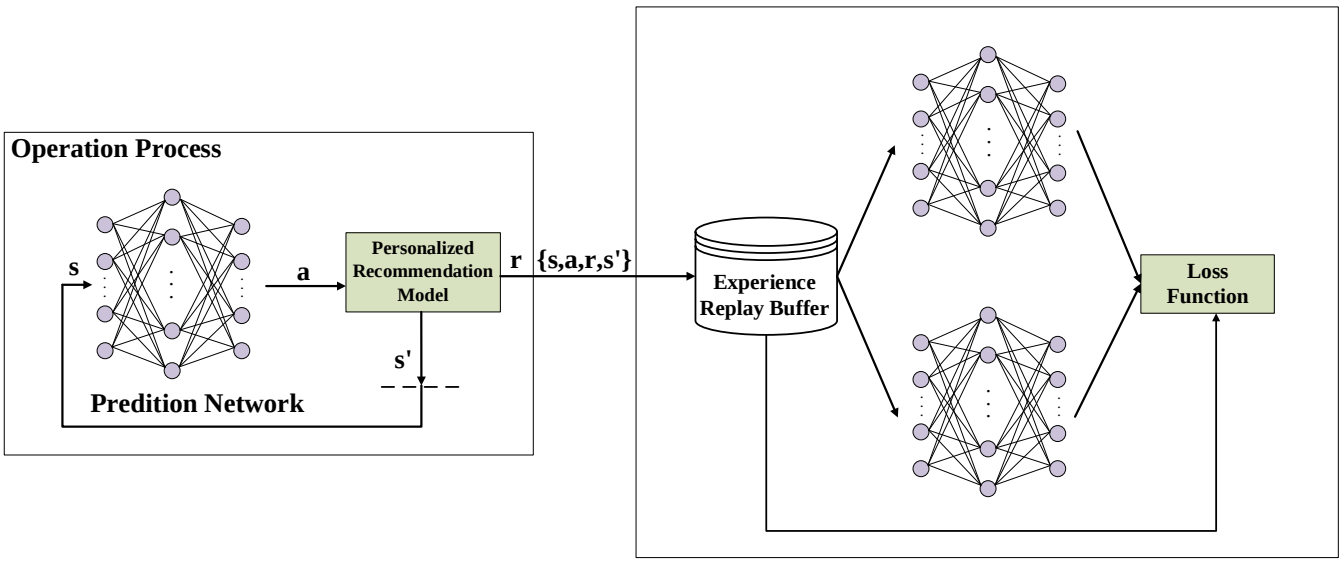


Fig. 6. Training process of the exercise record modification model

state $s_t$, which is calculated by the prediction Q network. $r_t + \gamma max_a Q_{\bar{\theta}}(s_{t+1}, a)$ represents the real value of the reward that can be obtained by choosing action $a_t$ in state $s_t$. $Q_{\bar{\theta}}(s_{t+1}, a)$ is calculated by the target Q network, indicating the maximum reward value that can be obtained in the next state $s_{t+1}$. $r_t$ is the current reward value that can be obtained by the reward function (8).

The gradient of the loss function is as (10). The network parameters are updated according to the gradient descent.

$$\frac{\partial L(\theta)}{\partial \theta} = \left(r_t + \gamma\, max_a\, Q_{\bar{\theta}}\,(s_{t+1}, a) - Q_\theta(s_t, a_t)\right) \frac{\partial Q_\theta(s_t, a_t)}{\partial \theta} \quad (10)$$

## IV. Experiment and Result

### A. Datasets

In this experiment, we utilized two real-world datasets of learner exercise records: ASSISTments09 and ASSISTments12. These datasets, widely used in educational research, contain authentic student response records collected from the ASSISTments online learning platform, developed by Worcester Polytechnic Institute in the United States. Details of the ASSISTments09 and ASSISTments12 datasets are provided in Table II.

TABLE II
Dataset description

| Dataset name | Number of learners | Number of exercises | Number of submitted records | Average number of submissions |
|---|---|---|---|---|
| ASSISTments 09 | 3840 | 26583 | 341357 | 88.89 |
| ASSISTments 12 | 42058 | 110512 | 6017201 | 143.07 |

### B. Experimental Analysis

We use Hit Ratio, Normalized Discounted Cumulative Gain, Mean Reciprocal Rank, and Mean Average Precision as metrics. The RLER model proposed in this paper is compared with three classical recommendation algorithms based on Collaborative Filtering and Matrix Decomposition (FM, MLP, and DeepFM), an algorithm based on Recurrent Neural Network (LSTM), and an algorithm based on attention mechanism (NAIS) [13].

The experimental analysis is conducted across the following three dimensions:

1) Performance of RLER model

Firstly, we verify the changes in recommendation performance of the RLER model proposed in this paper after considering learners' potential knowledge level and modifying their exercise records. Baseline is a personalized recommendation model that only considers learners' preferences. Baseline+dkt considers learners' potential knowledge level during recommendation. RLER model proposed not only considers learners' potential knowledge level, but also modifies learners' exercise records.

Figure 7-12 shows the comparison results of baseline, baseline+dkt, and RLER on the four metrics of HR, NDCG, MAP, and MRR.

As shown in Figure 7-9, in the dataset ASSISTments09, adding the learner's potential knowledge level to the learner's characteristics, HR increased by 2.5%-3.8%, NDCG increased by 1.7%-2.5%, and MAP@20 increased by 1.65%, MRR@20 increased by 1.24%. After modifying the exercise records, HR increased by 8.0%-9.8%, NDCG increased by 4.5%-5.6%, MAP@20 increased by 5.02%, and MRR@20 increased by 4.9%.

As shown in Figure 10-12, in the dataset ASSISTments12, adding the learner's potential knowledge level to the learner's characteristics, HR increased by 2.2%-3.4%, NDCG increased by 1.7%-2.4%, MAP@20 increased by 1.8%, and MRR@20 increased by 1.4%. After modifying the exercise records, HR increased by 9.5%-11.2%, NDCG increased by 5.4%-6.5%, MAP@20 increased by 6.4%, and MRR@20 increased by 5.6%.

In summary, we can draw three conclusions:

- Considering the learner's potential knowledge level when modeling the learner's characteristics can enhance the effectiveness of the recommendations.
- After using the reinforcement learning algorithm to modify the learner's exercise records, the recommendation effect has been greatly improved both in hit rate and sorting ability.
- Compared to dataset ASSISTments09, dataset ASSISTments12 contains more exercises and the average number of submissions by learners, indicating that learners are more likely to mistakenly select unsatisfactory exercises.

Therefore, after modifying the learner's exercise records, the recommendation effect is improved more significantly.

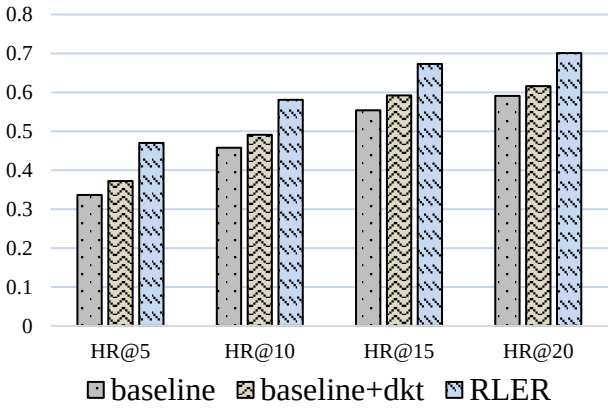


Fig. 7. HR of the data set ASSISTments09

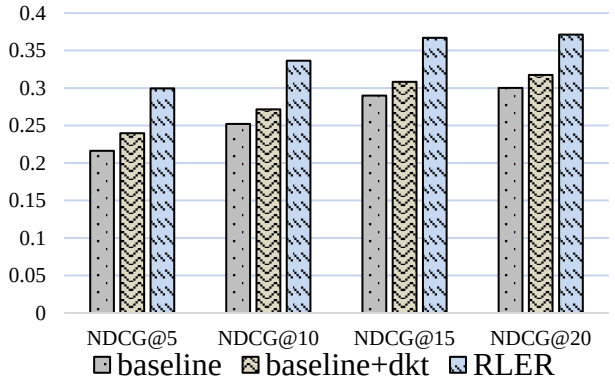


Fig. 8. NDCG of the data set ASSISTments09

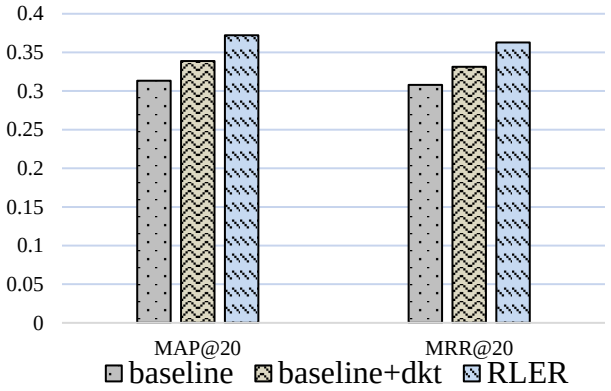


Fig. 9. MAP@20 and MRR@20 of the data set ASSISTments09

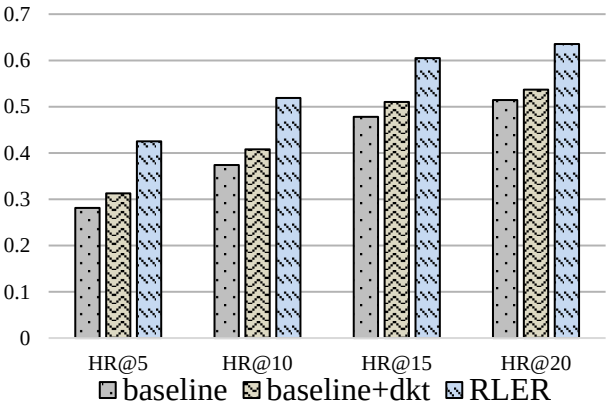


Fig. 10. HR of the data set ASSISTments12

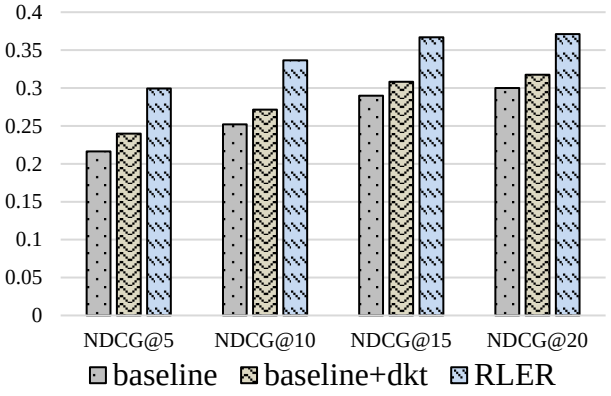


Fig. 11. NDCG of the data set ASSISTments12

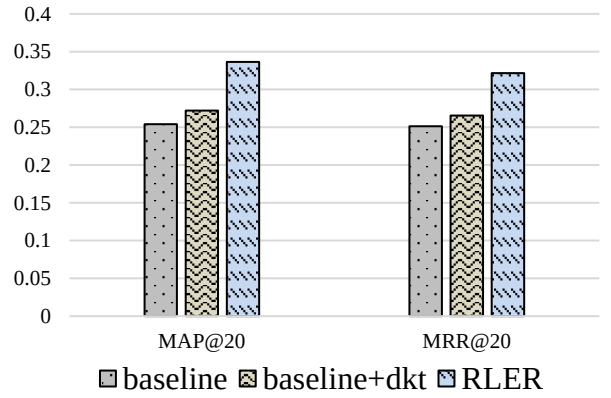


Fig. 12. MAP@20 and MRR@20 of the data set ASSISTments12

2) Comparison with other advanced methods

Secondly, we compare the RLER model with other personalized recommendation models.

As shown in Figure 13-15, in the dataset ASSISTments09, compared with other models, the RLER model increased by at least 7.5% for HR@5, 6.5% for HR@10, 3.8% for NDCG@5, and 4.4% for NDCG@10. MAP@20 has increased by at least 3.3%, and MRR@20 has increased by at least 3.0%. These data show that the RLER model outperforms other models both in accuracy and ranking performance in the dataset ASSISTments09.

As shown in Figure 16-18, in the dataset ASSISTments12, compared with other models, the RLER model increased by at least 8.5% for HR@5, 8.7% for HR@10, 5.2% for NDCG@5, and 5.6% for NDCG@10. MAP@20 has increased by at least 5.1%, and MRR@20 has increased by at least 4.6%. These data show that the RLER model outperforms other models both in accuracy and ranking performance in the dataset ASSISTments12.

In summary, we can draw four conclusions:

- The improved algorithms based on collaborative filtering and matrix decomposition generally do not perform well due to data sparsity problems in both datasets, with the FM model performing the worst.
- Although the LSTM model based on the recurrent neural network considers the temporal relationship, it does not account for the different importance of various exercises in the exercise records for the recommendation results, Although compared with FM, MLP and DeepFM, the experimental results are improved, the improvement is limited.
- NAIS considers the weight of different exercises for the recommended results, resulting in better performance. However, it still has drawbacks, such as the fact that it does not completely eliminate the effects of wrong choice exercises.
- The results of RLER are better than NAIS in all indicators, demonstrating the effectiveness of incorporating the learner's knowledge level in exercise recommendation and using reinforcement learning to modify the learner's exercise record. By comparing various indexes, our proposed method is superior to other methods.

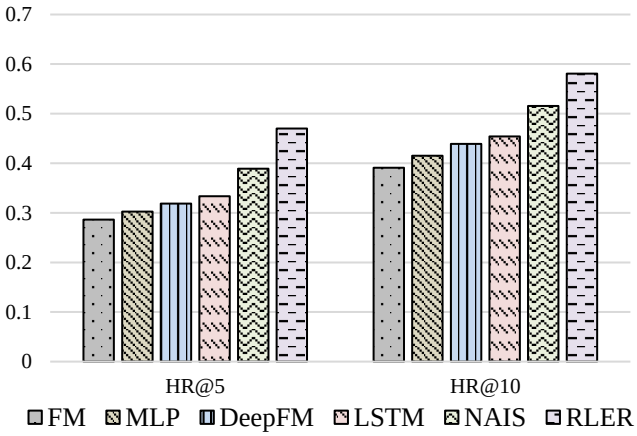


Fig. 13. HR@5 and HR@10 of the data set ASSISTments09

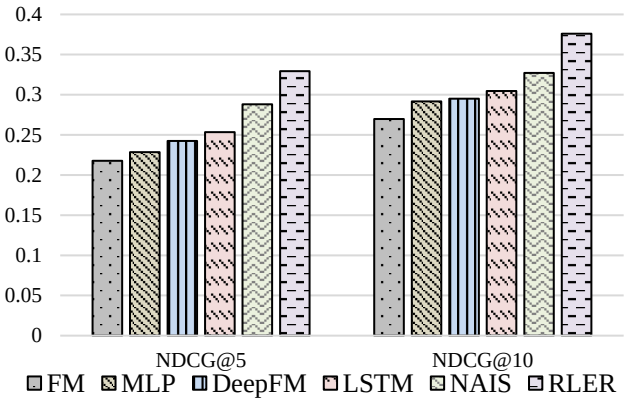


Fig. 14. NDCG@5 and NDCG@10 of the data set ASSISTments09

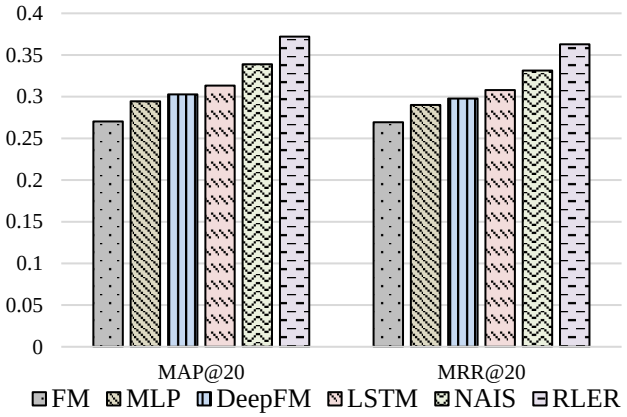


Fig. 15. MAP@20 and MRR@20 of the data set ASSISTments09

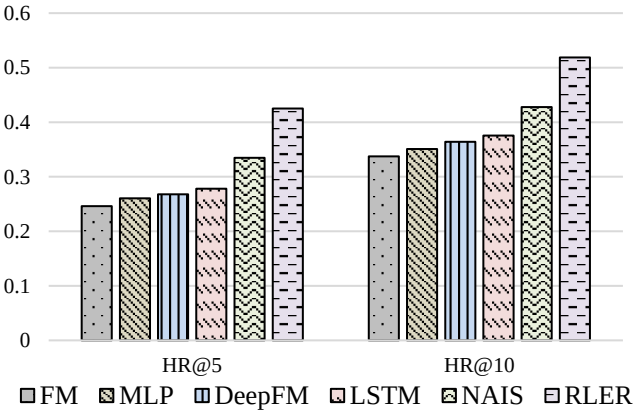


Fig. 16. HR@5 and HR@10 of the data set ASSISTments12

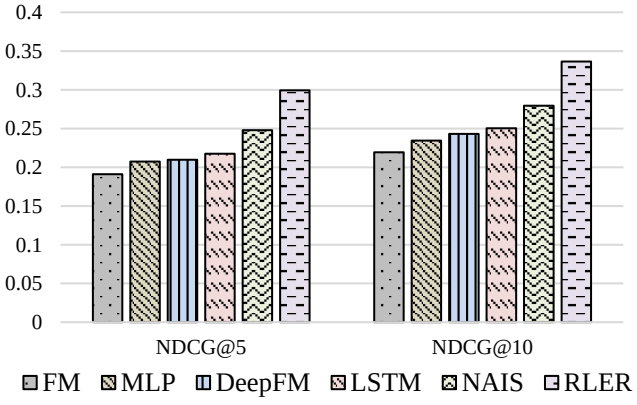


Fig. 17. NDCG@5 and NDCG@10 of the data set ASSISTments12

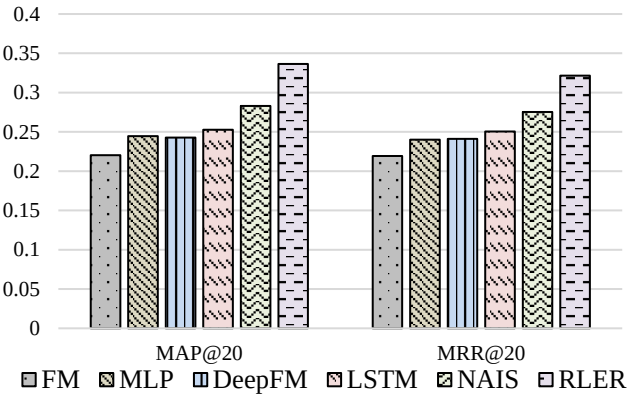


Fig. 18. MAP@20 and MRR@20 of the data set ASSISTments12

3) Universality of the exercise record modification algorithm

Finally, we verify whether the modified exercise record only improves the RLER model or has an effect on all recommended models. Figures 19-24 show that under two datasets, FM, MLP, DeepFM, LSTM, and NAIS models recommend based on original exercise records and modified exercise records on HR@10, NDCG@10, MAP@20.

In the dataset ASSISTments09, recommendations are made based on the modified exercise records, HR@10 Improved by 4.9%-5.7%, NDCG@10 Increased by 2.9%-3.7%, MAP@20 Improved by 3.0%-3.9%.

In the dataset ASSISTments12, recommended based on modified exercise records, HR@10 Increased by 6.4%-7.5%, NDCG@10 Improved by 4.2%-4.7%, MAP@20 Increased by 4.5%-5.1%.

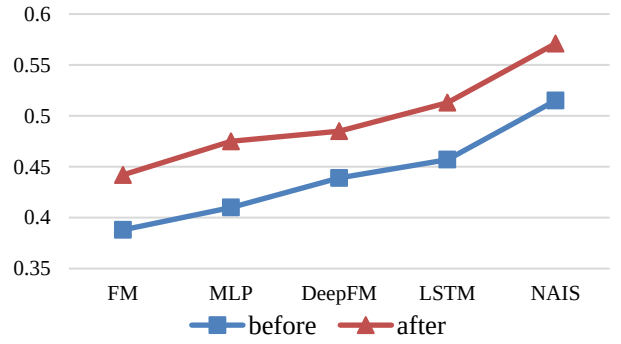


Fig. 19. HR@10 before and after the modification of the data set ASSISTments09

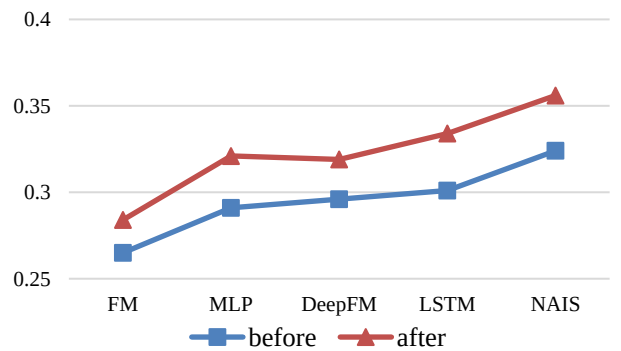


Fig. 20. NDCG@10 before and after the modification of the data set ASSISTments09

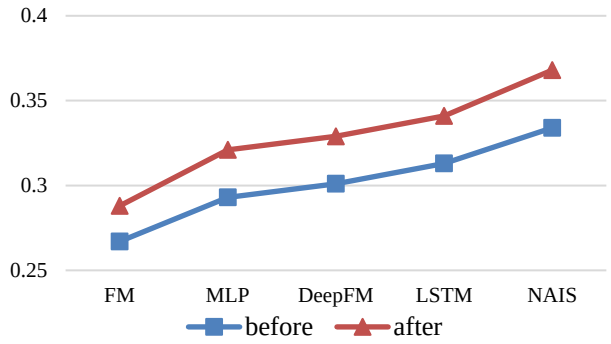


Fig. 21. MAP@20 before and after the modification of the data set ASSISTments09

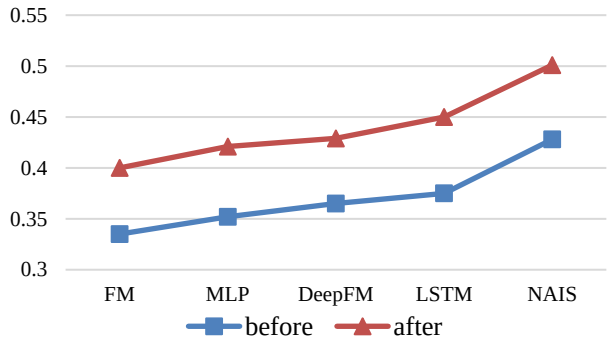


Fig. 22. HR@10 before and after the modification of the data set ASSISTments12

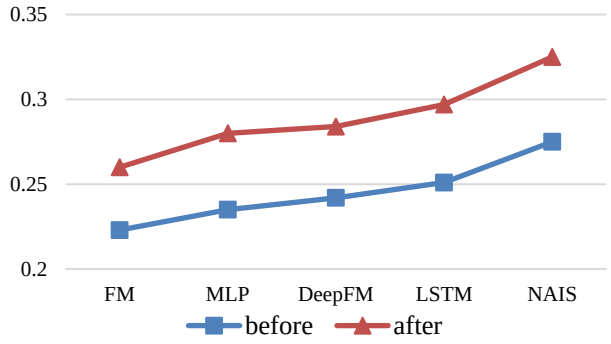


Fig. 23. NDCG@10 before and after the modification of the data set ASSISTments12

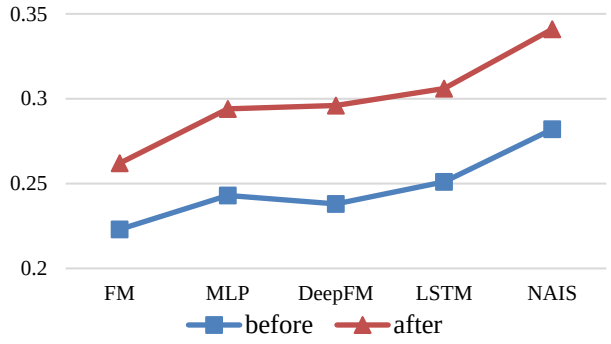


Fig. 24. MAP@20 before and after the modification of the data set ASSISTments12

In summary, we can draw two conclusions:

- Based on the modified exercise records, the HR and NDCG of all comparison models have improved, indicating that the learner's exercise record modification algorithm can be applied to any recommendation model.
- Since the exercise record modification algorithm is trained based on the personalized recommendation model proposed in this paper, the improvement effect on other recommendation models is not as significant as on RLER.

## V. Conclusion

In this paper, we propose an exercise recommendation method based on reinforcement learning DQN algorithm called RLER. Firstly, RLER utilizes the knowledge tracking model based on the long-short term memory network to assess the learner's potential knowledge level and incorporates it as part of the learner's characteristics. Subsequently, using the Deep Q Network algorithm, the exercise record modification model is designed to eliminate unsatisfactory exercises that learners mistakenly select during the learning process. Finally, based on the modified exercise record and learners' potential knowledge level, exercises are recommended for learners. Extensive experimental results demonstrate the effectiveness of RLER.

## References

[1] Goldberg D, Nichols D, Oki B M, et al. "Using collaborative filtering to weave an information tapestry," Communications of the ACM, 1992, 35(12): 61-70.

[2] Salehi M. "Application of implicit and explicit attribute based collaborative filtering and BIDE for learning resource recommendation," Data & Knowledge Engineering, 2013, 87: 130-145.

[3] Segal A, Katzir Z, Gal K, et al. "Edurank: A collaborative filtering approach to personalization in e-learning," Educational Data Mining 2014.

[4] Zhao J C, Lei C, Jian X G. "A recommendation algorithm based on collaborative filtering technology in distance learning," Electronic and Automation Control Conference (IAEAC), IEEE, 2017:2376-2379.

[5] Hudak G. "Labeling: Pedagogy and politics," Routledge, 2014.

[6] Dwivedi S, "Roshni V S K. Recommender system for big data in education," 2017 5th National Conference on E-Learning & E-Learning Technologies (ELELTECH). IEEE, 2017: 1-4.

[7] He X, Liao L, Zhang H, Nie L. "Neural Collaborative Filtering," International World Wide Web Conferences Steering Committee, 2017,31(2):7-16.

[8] Yudelson M V, Koedinger K R, Gordon G J. "Individualized bayesian knowledge tracing models," Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013. Proceedings 16. Springer Berlin Heidelberg, 2013: 171-180.

[9] Piech C, Spencer J, Huang J. "Deep Knowledge Tracing," Advances in Neural Information Processing Systems. 2015, (8):505-513.

[10] Kober J, Bagnell J A, Peters J. "Reinforcement learning in robotics: A survey," International Journal of Robotics Research, 2013, 32(11): 1238-1274.

[11] Wang X, Sandholm T. "Reinforcement learning to play an optimal Nash equilibrium in team Markov games," Advances in Neural Information Processing Systems, 2002, (15): 1571–1578.

[12] Rummery G A, Niranjan M. "On-Line Q-Learning Using Connectionist Systems." Cambridge, UK: University of Cambridge, Department of Engineering, 1994:112-118.

[13] Chua T S, He X, He Z, Liu, Z., Jiang, Y G, Chua T S. "NAIS: Neural Attentive Item Similarity Model for Recommendation," IEEE Transactions on Knowledge and Data Engineering, 2018:89-96.

**Simiao Yu** is a lecturer of software engineering at the School of Computer and Software Engineering, University of Science and Technology Liaoning. Her main research interests include: big data technology, artificial intelligence technology, etc.

**Ji Li** is currently pursuing Ph.D. degree at Northeastern University in Shenyang, China. He obtained B.S. degree in Information and Computing Science from Shenyang University of Technology in 2018 and a M.S. degree in Computer Software and Theory from Northeast University in Shenyang, China in 2021. His research direction is index and graph research.

**Tiancheng Zhang** is an associate professor at the Institute of Computer Software and Theory, Northeastern University. He is also a member of ACM and China Computer Society. His main research interests include: big data technology, data flow analysis and mining, spatio-temporal data management technology, artificial intelligence technology, etc.