International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 7986-7999

https://doi.org/10.70517/ijhsa463683

# Research on Automated Push System of Personalized Learning Content for Students Supported by Artificial Intelligence

**Xiaolong Zhou[1,2,*], Ming Liu[3] and Laidi Wu[4]**
[1] School of Marxism, Xinjiang Normal University, Urumqi, Xinjiang, 830017, China
[2] School of Law, Tongling University, Tongling, Anhui, 244061, China
[3] School of Mathematics and Computing, Tongling University, Tongling, Anhui, 244061, China
[4] Huangzhou High School of Hubei Province, Huanggang, Hubei, 438000, China
Corresponding authors: (e-mail: 252455@tlu.edu.cn).

**Abstract** Traditional teaching models often fail to adjust the learning content and progress according to the actual learning situation of each student. In order to solve this problem, this paper proposes a student learning content recommendation model based on personalized exploration strategy, which can automatically push appropriate learning resources according to students' personalized needs. By analyzing the interaction behavior of students' historical learning data and educational videos, a student personalized knowledge tracking model is designed and combined with LinUCB algorithm to realize the recommendation of educational videos. The experimental results show that on the POJ dataset, the model improves the accuracy by 1.05% and the AUC by 2.56% compared with the traditional model. On the LLS dataset, the MSE decreased by 4.11%. The model is able to effectively capture students' knowledge mastery status and recommend suitable learning videos based on their personalized characteristics. In addition, the model adopts parallel matrix computation with personalized exploration strategy to improve the computational efficiency and recommendation accuracy. The experimental results validate the potential of the system in the field of education, which can provide students with more personalized and intelligent learning support.

**Index Terms** personalized learning, LinUCB, knowledge tracking, educational videos, recommender system, personalized exploration strategy

## I. Introduction

With the continuous progress of science and technology, the application of artificial intelligence technology in the field of education has become increasingly popular [1], [2]. As the main body of education, the problems and challenges encountered by students in their learning process have become diverse [3]. In order to help college students learn and master knowledge better, it is of great significance to study the automated push system of students' personalized learning content based on artificial intelligence to improve the quality of teaching [4], [5].

The automated push system based on students' personalized learning content uses machine learning, natural language processing, and other technologies, aiming to provide personalized learning support and guidance by integrating artificial intelligence technologies in order to realize intelligent learning tutoring [6]-[8]. The design goal of the system is to meet the personalized learning needs of college students, provide an efficient and convenient learning experience, and enhance students' learning outcomes, with the functions of recommending learning resources suitable for students, intelligent Q&A answers, automatic homework correction, and learning progress management to help students learn better and enhance their learning outcomes [9]-[12]. In the implementation process, the design and realization of data collection and pre-processing, intelligent recommendation system, natural language processing and Q&A system, and homework correction system need to be fully considered [13]-[15]. Through the evaluation and improvement of the system effect, the degree of intelligence and user experience of the system can be continuously optimized to provide better support and assistance for college students' learning [16], [17].

With the rapid development of information technology, the demand for personalized learning is gradually growing. In the traditional education system, the teaching content and progress are usually uniform, and this model does not fully consider the individual differences and learning progress of students. The core concept of personalized learning is to provide tailor-made learning content according to each student's learning situation, interests and learning habits, in order to achieve a higher learning effect. However, how to accurately and effectively push personalized learning content based on students' knowledge mastery remains a challenge in current educational technology research.

This study aims to design a personalized recommendation system based on students' learning history data. The system pushes the video resources most suitable for students' current learning status through real-time tracking of students' knowledge mastery status and combining with recommendation algorithms for educational videos. In order to achieve this goal, a student knowledge tracking model is first constructed, which can accurately predict students' mastery on each knowledge point. Through the LSTM network, the model is able to predict students' mastery of knowledge points in the future based on their historical answer records and interactive behaviors. Secondly, the LinUCB algorithm is combined to recommend educational video resources, and real-time adjustments are made using students' feedback on different learning resources. In the research process, multiple datasets are used for validation to ensure the effectiveness and universality of the model. Through comparative experiments, the personalized learning recommendation system proposed in this paper outperforms the traditional model in terms of accuracy and recommendation effect. This result provides strong technical support for the automated push of personalized learning content.

## II. Student learning content recommendation model based on personalized exploration strategy

In order to realize the design of automated push system for students' personalized learning content, this paper materializes the learning content into educational videos, and carries out knowledge tracking for students, and then conducts feature extraction for teaching videos, and proposes LinUCB, a recommendation model for teaching video resources based on personalized exploration strategy.

### *II. A. Student Personalized Knowledge Tracking Model*

#### II. A. 1) Context-based Q-matrix representation

Contextual representation of the $Q$ matrix through the historical student interaction response matrix and the $Q$ matrix. For the historical data of student interactions, Knowledge Point $\{c_1, c_2, ..., c_k\} \in C$ , Exercise $\{e1, e2, ..., e_M\} \in E$ , Student $\{s_1, s_2, ...s_N\} \in S$ . student's historical interactions $R \in R^{N \times M}, r \in \{0,1\}$ . student answered exercises are represented as $e(s) = \{e(s_1), e(s_2), ..., e(s_N)\}$ .

In the original $Q$ matrix, the rows represent a test item and the columns represent the basic concepts or knowledge points (KCs) examined in that test item. If an item $kc:k$ is examined in Exercise $i$ , $q_{ik}$ it is 1, otherwise it is 0. For each quiz item, $Q \in R^{M \times K}$ it is no longer possible to use 0 or 1 to indicate the association of a KC with an exercise because the importance of each KC may be different in an exercise, and therefore different weights are obtained by calculating the frequency with which each KC is examined in all exercises. Generally, it can be assumed that the higher the frequency of a knowledge point, indicating its greater importance, the greater the weight. The calculation of knowledge point weights is shown in equation (1).

$$w(k_i) = \frac{\sum_{r=0}^{M} Q_{ri}}{\sum_{r=0}^{M} \sum_{l=0}^{N} Q_{rl}} \tag{1}$$

where $w(k_i)$ denotes the weight of the $i$ nd knowledge point among all the knowledge points, $N$ is the column of the matrix and $M$ is the row of the matrix. Weight the $Q$ matrix according to the weight coefficients of the knowledge points to calculate equation (2), and for each row of any exercise $e_j$ , each non-zero value maps to a decimal between 0 and 1, and the weight values add together to 1. i.e:

$$R_m(k_i) = \frac{Q_{mi}}{\sum_{j=1}^{K} Q_{kj}} \tag{2}$$

Student performance on knowledge points is calculated by recording student answer interactions. The overall level of student mastery on a knowledge point varies as students practice. The complex learning scenarios of the student learning process include factors such as the student's learning background, the difficulty of the quiz items, and the correlation between the exercises. Contextual factors for knowledge points across the practice space are obtained by basing the data on the history of student practice interactions. For each knowledge point, $\delta$ is used to represent the context factor of knowledge point $l$, as shown in equation (3):

$$\delta_i = \begin{cases} 0, & (q_{jl} = 0) \\ \dfrac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij} q_{jl}, & otherwise \end{cases} \tag{3}$$

where $q_{jl}$ is the weight of knowledge point $l$ on quiz item $j$, and $r_{ij}$ denotes the student $i$ response result on exercise $j$. The context factor $\delta$ on each knowledge point is obtained by calculating the results of students' responses on all exercise spaces for that knowledge point.

The values in the weighted $Q$ matrix are scaled by the context factors of the knowledge points to finally obtain the contextualized $CQ$ matrix equation (4), where each row represents an exercise represented by $KC$ and each column represents the presence of the knowledge point in the exercise:

$$CQ[:,l] = \delta_l \times Q[:,l] \tag{4}$$

**II. A. 2) Prediction of students' state of knowledge acquisition**

In traditional deep knowledge tracking models, students' historical answer sequences are input via one-hot coding, and the coded sequences are of length 2M for a practice sequence of M exercises. the input vectors are not only highly sparse, but also contain only limited semantic information. In the contextualized $Q$ matrix, each column represents a specific knowledge point vector and is combined with the student's performance in past exercises. The feature vector of the lth knowledge point can be represented as $q_{1l}, q_{2l}, ..., q_{El}$.

The feature vectors $q_t$ of the knowledge points and the students' answer results $r_t$ are used as input sequences to the knowledge tracking model $(q_t, r_t)$. where each input sequence is of length $k+1$. In the knowledge tracking model based on the context of the students' answer data, the focus is on the performance of the students on each knowledge point, which reduces the dimensionality of the model's input data and accelerates the convergence of the model.

The output of the LSTM network [18] is a vector of probabilities $P(K^m)$ predicting that each knowledge point can be answered correctly, with a length equal to the number of knowledge points:

$$P(K^m) = [P(k_1^m), P(k_2^m), ..., P(k_n^m)] \tag{5}$$

And the deviation between the predicted results and the actual observed results is minimized by a binary cross-entropy loss function, which is trained by a stochastic gradient descent SGD optimizer:

$$J(\theta) = -\frac{1}{(T+1)}\sum_{t=0}^{T}(r_t \log(\hat{p}_t) + (1-r_t)\log(1-\hat{p}_t)) \tag{6}$$

**II. A. 3) Projected scope of knowledge**

The probability of knowledge points appearing in the next time step is predicted by the historical sequence of knowledge points examined in the student's history of doing questions. Therefore, an LSTM network can be used to predict the range of knowledge points that the user will study in the next time step, using the softmax function as the activation function of the output layer:

$$\{kc_1 \mid P(K^{c1}), kc_2 \mid P(K^{c2}), kc_3 \mid P(K^{c3}), ...., kc_n \mid P(K^{cn})\} \tag{7}$$

The input sequence of knowledge points in the model is obtained from the students' historical answer sequences and encoded with one-hot coding, defined as $K_t$. It denotes the prediction of the knowledge points that appear at the next time step. In LSTM network, the optimized loss function for a single student can be expressed as:

$$L_{S(C)} = \sum_{t=0}^{T} l_b(Z^t . \phi(k^{t+1}), 1) \tag{8}$$

where the symbol $(\cdot)$ denotes the dot product operation, $l_b$ denotes the binary cross-entropy loss function, and the output vector $P(K_i^c)$ of the LSTM model, the length of which is the number of knowledge points contained in the set of exercises, and each element denotes the probability of the corresponding knowledge point appearing at the $t+1$ moment.

Through LSTM we get the probability of each knowledge point appearing in the next moment of the student's learning, and in the exercise recommendation, we need to recommend exercises for the students that can cover the knowledge points that the students are learning to improve the students' learning efficiency. At the same time, in order to improve the novelty of the recommendation, and combined with the learning characteristics of the actual learning process of students. For students to answer the question process, the number of visits is relatively high, and the user answer the question with a high rate of correctness, that is, the user has been able to better master the part, to reduce the number of times it occurs, to reduce the ineffective practice. For the knowledge points that are poorly mastered by the user or examined less frequently, it is necessary to increase their weights and increase the probability of their occurrence. Therefore, in the output layer of LSTM, add a weight vector, denoted as:

$$w(K) = [w(k_1), w(k_2), \ldots. w(K_m)] \tag{9}$$

where the length of $m$ is equal to the number of knowledge points. $w(k_i)$ can be expressed as:

$$W(K_i) = \begin{cases} 1 - \dfrac{r_i}{c_i} & c_i > 0 \\ 1 & c_i = 0 \end{cases} \tag{10}$$

where, $c_i$ is the number of occurrences of the knowledge point, and $r_i$ denotes the number of times the knowledge point was answered correctly, so the knowledge points that were examined less frequently and with a low rate of correctness were increased with their corresponding weights to finally obtain the desired probability of the knowledge point's occurrence in the next exercise:

$$P(K^c) = P(K_i^c).w(k_i) \tag{11}$$

***II. B.Educational Video Feature Extraction***

In online education scenarios, it is usually impossible to obtain information involving learners' privacy, such as gender, age, and educational background. Based on this, this paper utilizes the historical learning behavior records of learners in the preprocessed dataset for feature extraction. That is, based on the learning behavior fields of all learners, a total of five representative and important attributes of educational videos, namely, learning rate, average completion, replay count rate, correctness, and difficulty, are extracted as inputs to the LinUCB [19] recommendation model.

The calculation steps for the first four educational video feature values are as follows:

$$fracstudy_j = \frac{M_j}{total\_num\_std} \tag{12}$$

$$avgfracComp_j = \frac{\sum_{i=1}^{M_j} fracComp_{ij}}{M_j} \tag{13}$$

$$fracnumRWs_j = \frac{\sum_{i=1}^{M_j} numRWs_{ij}}{M_j^{*} numRWs_{max}} \tag{14}$$

$$fracCorrect_j = \frac{\sum_{i=1}^{M_j} success_{ij}}{M_j} \tag{15}$$

where subscript $j$ represents the $j$ nd educational video, $M_j$ represents the number of learners participating in video $v_j$, $total\_num\_std$ is the total number of learners. $fracComp_i$ represents the degree of completion of the video $v_j$ by the learner $l_i$. $numRWs_{ij}$ represents the number of times the learner $l_i$ plays back the video $v_j$

while watching it, and $mumRWs_{\max}$ is a threshold for the number of times the video is played back. $success_{ij}$ indicates whether the learner $l_i$ completed the accompanying test correctly after learning the video $v_j$.

Finally, in order to ensure the adaptability of the educational video in online learning, this paper determines the difficulty of the educational video through the learner's learning behavior records, and puts forward a linear hypothesis: the difficulty of the educational video is linearly negatively correlated with the average degree of completion of the learner who watched the video, and the correctness rate of the accompanying quiz after watching the video, and linearly positively correlated with the rate of the number of times a learner played back when he or she watched the video. The difficulty characteristics of the educational videos are then calculated as follows:

$$dif_j = m_1 * (1 - avgfracComp_j) + m_2^{*}(1 - fracCorrect_j) + m_3 * fracnumRWs_j \tag{16}$$

where $dif_j$ denotes the difficulty of the video $v_j$, with values ranging from 0 to 1. $m_1, m_2, m_3$ is the importance weight that controls the three factors associated with the difficulty calculation, and $m_1 + m_2 + m_3 = 1$. As a result, as long as a learner interacts with an educational video resource and generates a record of the video viewed, the five attributes of that educational video can be regarded as a vector of behavioral features of the learner. Finally, each learner generates multiple learning records, and reading the educational video contained in that record in turn can all form a behavioral feature matrix.

### *II. C.Educational Video Resource Recommendation Model Based on LinUCB Algorithm*

In online education scenarios, when a learner makes a request to an online learning platform, the recommendation engine therein will retrieve the learner's historical learning behavior records and recommend suitable educational videos for him based on his learning ability characteristics. In this paper, it is assumed that the learner's learning content and interest will not change in a short period of time, so the time problem is not considered and the time iteration of traditional LinUCB is omitted. Compared with the existing LinUCB, the algorithm has two aspects of improvement, focusing on the recommendation problem from the overall and individual perspectives, respectively.A comparison of the characteristics of the LinUCB algorithm before and after the improvement is shown in Fig. 1.

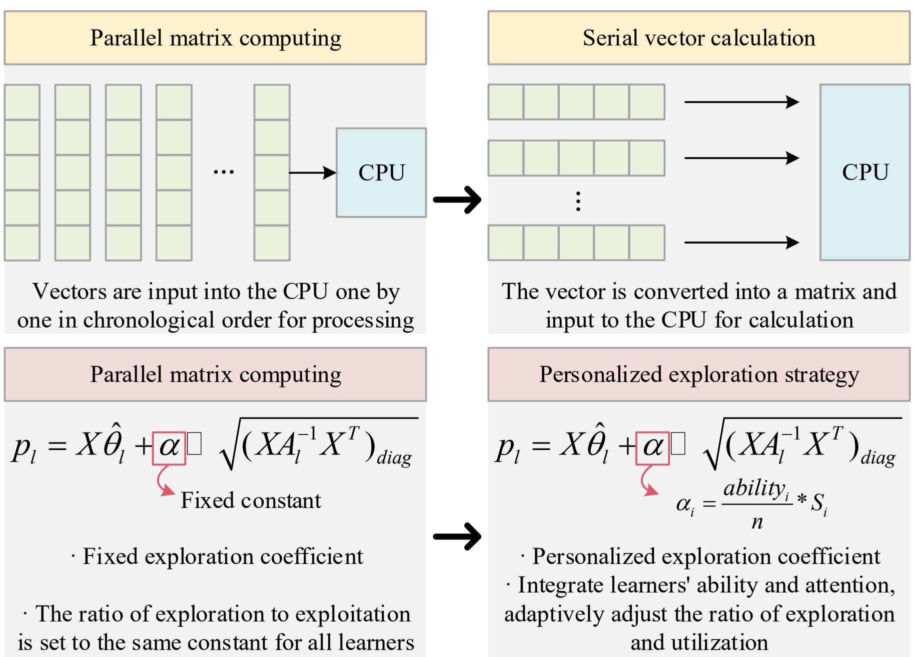



Figure 1: Comparison of the LinUCB algorithm before and after improvement

#### II. C. 1) Parallel matrix calculation

In order to increase the computational speed and reduce the usage cost, we combine multiple serial vectors of the traditional LinUCB into one parallel matrix for computation.LinUCB assumes that the expected value of the feedback gain of each arm $r$ is a linear function of the eigenvector $x$ of each user. In the online education scenario of this paper, the arms in LinUCB refer to all the educational video resources provided by the online learning platform, and the users are the learners who participate in the learning of the educational video resources. The expected returns of all the arms in the improved LinUCB are calculated as follows:

$$E[r_l \mid X] = X\hat{\theta}_l \tag{17}$$

where $r_l = [r_l^1, \cdots, r_l^j, \cdots, r_l^n]^{\bullet}$ . $r_i^j$ is the expected value of the feedback benefit of the learner $l$ learning the $j$ th educational video in the set of educational video resources $c$, and $n$ denotes the number of educational videos in the set of online learning resources $c$. $X = [x_1^{\bullet}, \cdots, x_j^{\bullet}, \cdots, x_n^{\bullet}]^{\bullet}$ is a $n \times d$ -dimensional matrix where $x_j^{\bullet}$ is the feature vector of the $j$ th video in the educational video resource set $C$ , $x_j^{\bullet} = [fracstudy_j, avgfracComp_j, fracnumRWs_j, fraccorrect_j, dif_j]$. $d$ denotes the number of video features, and here $d = 5$. $\hat{\theta}_i = [\hat{\theta}_i^1, \cdots, \hat{\theta}_{i_i}^k, \cdots, \hat{\theta}_i^d]^{\bullet}$ is a $d \times 1$ -dimensional vector of coefficients to be learned for obtaining the rewards of each dimensional feature, and its row elements $\hat{\theta}_i^k$ refer to the parameters of the $k$ th video feature. $\hat{\theta}_i$ is computed as follows:

$$\hat{\theta}_l = (D_l^{\bullet} D_l + I_d)^{-1} D_l^{\bullet} c_l = A_l^{-1} b_l \tag{18}$$

$$\begin{cases} A_l = D_l^{\bullet} D_l + I_d \\ b_l = D_l^{\bullet} c_l \end{cases} \tag{19}$$

where $D_i = [x_1^{\bullet}, \ldots, x_i^{\bullet}, \cdots, x_{m_j}^{\bullet}]^{\bullet}$ is a feature matrix consisting of $m_i$ row vectors $x_i^{\bullet}$ of dimension $m_i \times d$. $x_i^{\bullet}$ denotes the feature vector of the $i$ th educational video in set $M_l$. All the educational videos contained in the behavioral records of learner $l$ form set $M_l$ with number $m_l$. order $c_l = [r_1, \ldots, r_i, \ldots, r_{m_l}]^{\bullet}$ , whose row element $r_i$ denotes the reward value of the $i$ th educational video in set $M_l$, is defined as the degree of completion of the learner $l$ with respect to the educational video. $A_i$ is a diagonal matrix of dimension $d \times d$ computed from the feature matrix $D_i$. $b_i$ is a vector of dimension $d \times 1$, and the row elements of $b_i$ represent the cumulative benefits obtained for each educational video feature.

With LinUCB's parallel matrix computation, we can obtain the total estimated reward of learner $l$ for the educational video:

$$p_l = X\hat{\theta}_l + \alpha_l \square \sqrt{(XA_l^{-1}X^{\bullet})_{diag}} \tag{20}$$

where $p_i$ is a $n \times 1$ vector whose row elements represent the total estimated reward for each video in $C$. $\alpha_l$ is also a vector of $n \times 1$ which controls whether the algorithm's decisions favor exploitation or exploration. The smaller $\alpha_l$ is, the higher the probability of selecting a recommended video from the watched videos. Conversely, the larger $\alpha_l$ is, the more likely it is to recommend new unwatched videos for the learner $l$. $(XA_i^{-1}X^{\bullet})_{diag}$ is a $n \times 1$ vector whose row elements are the diagonal elements of $XA_s^{-1}X^{\bullet}$ which represents the predicted variance of the expected returns $X\hat{\theta}_l$. The Hadamard product of $\alpha_l$ and $\sqrt{(XA_l^{-1}X^{\bullet})_{diag}}$ can be interpreted as the uncertainty of the $X\hat{\theta}_l$

### II. C. 2) Personalized Exploration Strategies

In order to adapt to the variation of different learners' abilities and perceive their acceptance of educational video resources of different difficulties, this paper adopts the attention mechanism to calculate the parameter $\alpha_l$ that can control the ratio of exploration and utilization. $\alpha_l$ is denoted as the personalized exploration coefficient, which is related to the learners' abilities and attention, and can reflect the personalized needs of different learners. It is defined as follows:

$$\alpha_l = \frac{ability_l}{n} * S_l \tag{21}$$

where $ability_i$ denotes the competence of the learner $l$. The more capable the learner is, the greater the need for exploration. $S_l$ is a $n \times 1$ vector representing the learner's attention to each video. By using the attention mechanism, $S_l$ is computed as follows:

$$S_l = D_{n \times m_s} * c_l \tag{22}$$

where, $D_{n \times m_i}$ is the difference degree matrix of $n \times m_l$, whose matrix elements represent the difference degree of difficulty between the videos in $C$ and $M_l$. The elements of row $p$ and column $q$ in $D_{n \times m_i}$ indicate that the Euclidean distance between the difficulty values of the $p$ th video in $C$ and the $q$ th video in $M_l$ can be calculated by the following equation:

$$D_{n \times m_i}[p,q] = \sqrt{(dif_{p \in C} - dif_{q \in M_i})^2} \tag{23}$$

where $D_{n \times m_i}$ sets the difficulty difference value between the videos that the learner has watched and the videos in $C$. $c_l$ performs a weighted summation of the row elements in $D_{n \times m_i}$ to obtain the learner's attention score for each educational video. As a result, $S_l$ decreases the level of attention of the learner $l$ for the watched videos and increases the weight of attention for the unwatched videos, and the attention score is calculated as shown in Schematic 2. When the value of the difficulty difference between the watched and unwatched educational video resources is larger, the more learners tend to learn these unwatched video resources, and the higher the attention score is, because they stimulate the learners' desire to explore the uncharted territory.

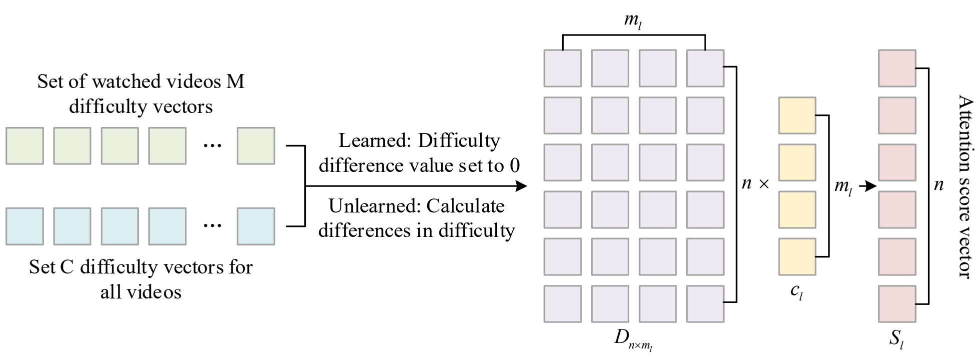



Figure 2: Calculation of learners' attention scores

For each learner, $ability_l$ and $S_l$ are different. Therefore, on the one hand, we design a personalized exploration strategy that can make full use of the learner's learning state and adaptively adjust the exploration ratio according to the learner's ability and attention. On the other hand, personalization of $\alpha_i$ ensures that the difficulty of the explored educational videos is within the learner's ability, thus reducing the risk of exploration.

Finally, the top $N$ videos with the highest reward values are selected from the row elements of the total estimated reward vector $p_l$ to form a recommendation list:

$$list[v] = top_N(p_{l,v}), p_l = [p_{l,v}] \tag{24}$$

where $p_{i,v}$ is an element in vector $p_l$ representing the reward value of video $v$ for learner $l$.

## III. Model application experiment and result analysis

This chapter provides an experimental investigation of the practical effects of the proposed student personalized knowledge tracking model and the educational video recommendation model, thus laying the foundation for the design of the recommender system.

### *III. A. Validation of the effectiveness of students' personalized knowledge tracking*

#### III. A. 1) Data sets and assessment indicators

(1) Datasets

This section uses the POJ dataset and the LLS dataset for the evaluation of students' personalized knowledge tracking models.

The POJ dataset is crawled from the Peking University online assessment platform, which is used for learners' programming exercises. In this paper, data preprocessing is first performed to remove learners who have attempted to submit the exercises less than twice and those who have been interacted with less than twice. Then, a total of 12 anomalous learners who submitted the exercises more than 6000 times are further removed. The final dataset contains 14,352 learners, 2,146 practice problems, and 435,126 interactions recorded for the time period between 2021-08-05 and 2022-05-14.

The LLS dataset was collected from an AI-driven EdTech company with a time period between 2021-11-20 and 2022-03-30, and contains both behavioral and personality information of learners. In this case, the learner's behavioral information is obtained from the company's online learning app, and the learner's personality information is obtained from the learning platform by inviting the learner to fill out a TIPI questionnaire. By filtering out the questionnaires with invalid responses, 2174 learners, 1205 courses and corresponding 324,583 interactions are finally retained, and each interaction contains the corresponding course ID, course completion time and course completion accuracy.

(2) Evaluation Metrics

In order to measure the effectiveness of the personalized knowledge tracking model, this section uses the accuracy (ACC) metrics and the area under the ROC curve size (AUC) metrics on the classification problem on the POJ dataset, and the mean square error (MSE) metrics on the regression problem on the LLS dataset, which are computed using the following formulas:

$$ACC = \frac{N_{correct}}{N_{total}} \tag{25}$$

$$MSE = \frac{1}{M}\sum_{u}\left(y_{aux} - \hat{y}_{aux}\right)^2 \tag{26}$$

where $ACC$ represents the number of samples correctly predicted by the model $N_{correct}$ as a percentage of the total number of samples $N_{total}$ predicted. $M$ is the number of learners in the test set, and $y_{aux}$ and $\hat{y}_{aux}$ represent the true and predicted accuracy of the course, respectively.

In addition, in order to make a more intuitive comparison, this paper uses the percentage of improvement (IP) to measure the percentage of improvement of the proposed method in this paper compared to the compared methods:

$$IP = \frac{METRIC_{our\_method} - METRIC_{compared\_method}}{METRIC_{compared\_method}} \tag{27}$$

where $METRIC$ stands for the relevant indicator such as ACC, AUC, MSE.

#### III. A. 2) Experimental results and analysis

(1) Model comparison experiment

In order to evaluate the performance of the proposed personalized knowledge tracking model, this paper compares it with two traditional models, BKT and BKT+Forget, as well as two deep learning neural network-based models, DKT and DKVMN. When dividing the dataset, this paper divides the learners and their corresponding learning behavior sequences into training, validation and testing sets in an 8:1:1 manner, respectively.

The effect of different methods on POJ and LLS datasets regarding ACC, AUC metrics and MSE metrics is shown in Table 1. Observing the data, it can be seen that BKT+Forget outperforms the BKT method among the traditional methods, where it improves 2.84% and 1.23% on the POJ dataset regarding ACC and AUC metrics, respectively, and reduces 1.43% on the LLS dataset regarding MSE metrics, and the result suggests that considering forgetting behaviors in the knowledge tracking task can help to improve tracking effectiveness.

In addition, the deep learning-based knowledge tracking models DKT, DKVMN [20], and the methods in this paper outperform the traditional methods BKT and BKT+Forget because, on the one hand, the deep learning-based methods are better able to represent the associations between the knowledge points, and on the other hand, the long and short-term memory networks used in the deep learning-based methods are more suitable for modeling sequential data, and thus can more accurately model the learner's behavior thus capturing the learner's knowledge level more accurately than traditional models.

Finally, among the deep learning based methods, the method proposed in this paper has the best performance on all metrics. On the POJ dataset, this paper's method improves the performance of the best performing baseline method DKT on ACC and AUC metrics by 1.05% and 2.56%, respectively. On the LLS dataset, the performance of this paper's method on MSE metrics is improved by 4.11% compared to DKT. This indicates that the method in this paper can well track the students' knowledge level effectively and has better efficacy compared to the existing methods.

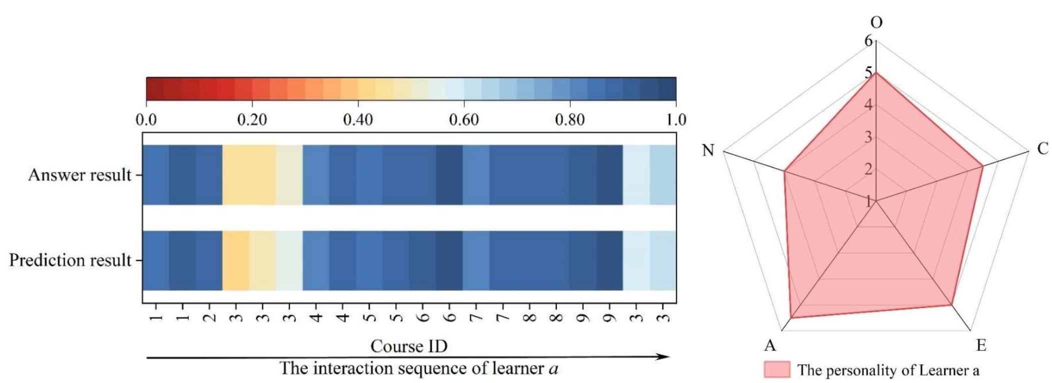
Table 1: Comparison of model knowledge tracking effect

| | | POJ | | LLS |
|---|---|---|---|---|
| | Model | ACC | AUC | MSE |
| Traditional methods | BKT | 0.6539 | 0.6524 | 0.1886 |
| | BKT+Forget | 0.6725 | 0.6604 | 0.1859 |
| Methods based deep learning | DKT | 0.6943 | 0.6687 | 0.1824 |
| | DKVMN | 0.6932 | 0.6613 | 0.1836 |
| | Our Method | 0.7016 | 0.6858 | 0.1749 |

(2) Case Study

To verify the effectiveness of the method in this paper even further, learner a on the LLS dataset is selected as a case for personalized knowledge tracking. The results of personalized knowledge tracking for learner a are shown in Figure 3. The mapping relationship between course IDs and courses is presented in Fig. (a), and course IDs 1~9 represent new, comprehensive, speaking, grammar, listening, vocabulary, practice, reinforcement, and review courses, respectively. The prediction result is the knowledge level result of learner a predicted by the personalized knowledge tracking model, and the answer result is the real answer result of learner a in the learning process. Figure (b) shows the radar chart of the five personality scores of learner a's Openness to Experience, Dutifulness, Extraversion, Likability, and Neuroticism, which are denoted by O, C, E, A, and N, with dimension values of 5, 4.5, 5, 5.5, and 4, respectively.

It can be seen that the personalized knowledge tracking model in this paper can model learners' individual differences well. Although learner a has high values in dimension E, and has good results in the new lesson, comprehensive lesson, grammar lesson, listening lesson, vocabulary lesson, practice lesson, reinforcement lesson, and review lesson, and has good English learning performance in general, it can be seen that this learner lacks in oral English, and the model can accurately capture this aspect of the learner's characteristics.




(a) The mapping relationship between course ID and courses (b) The personality of Learner *a*

Figure 3: Personalized knowledge Tracking Results of Learner *a*

### *III. B. Experimental analysis of model recommendation effect*

In this section, experiments are conducted to demonstrate the effectiveness of the model based on deep knowledge tracking for recommending educational video resources. The experiments are carried out in the following two aspects: evaluation of the optimization strategy and visual analysis of the recommendation effect, respectively.

#### III. B. 1) Optimization Strategy Evaluation

The evaluation metric used in the recommendation model is the cumulative reward value obtained after taking a series of recommendation actions, which reflects the level of the student's hidden knowledge state, with higher environmental rewards expressing a better recommendation strategy. Since the trained neural network uses random parameters during each initialization process to simulate different students' knowledge states. The variation of cumulative reward values with increasing number of training steps when recommending 8 educational videos sequentially from 30 topics is shown in Fig. 4. In the experiment, the Adam optimizer is chosen to update the parameters, the learning rate is set to 0.001, and the training batch size is 128. Since the model is constantly updated to find a better recommendation strategy during training, the cumulative reward value changes with the number of iterations, and the overall trend is upward.

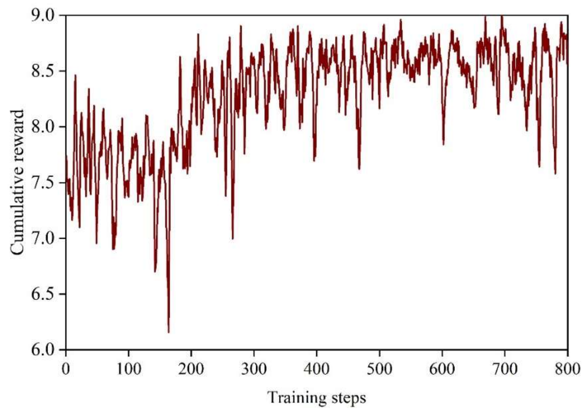



Figure 4: The cumulative reward value changes as the number of training steps increases

The loss function of the model also changes with the increase in the number of training steps, with an overall decreasing trend, and no longer decays after the optimal strategy is obtained. In order to better optimize the neural network, the loss function is normalized, after processing the data distribution between 0 and 1 the average value of the loss function with the change in the number of training steps is shown in Figure 5, which converges in a shorter period of time, and faster convergence speed can help the model to adapt to the environmental changes quickly.

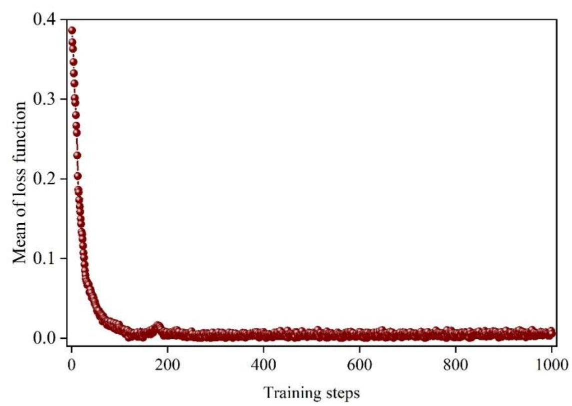



Figure 5: The variation of the mean value of the loss function

#### III. B. 2) Recommendation effect analysis

In order to visualize the recommendation effect of the model, this subsection will reflect the change of a certain student's knowledge state with the learning of different educational videos. Specifically, firstly, 12 consecutive videos are taken as samples from 100 recommended educational videos, the information of which is shown as horizontal coordinates in Fig. 6. Take the first information (300,45,1) as an example: 300 denotes the id of the topic, 45 denotes the id of the knowledge point, and 1 denotes the predicted learning status of mastered. These 12 educational videos cover a total of 7 knowledge points, and as the recommendation proceeds, the change in the student's predicted knowledge status for these 7 knowledge points is shown in Figure 6. Where darker colors represent lower mastery of a knowledge point by the student and lighter colors represent higher mastery of a knowledge point by the student.

The following results can be obtained after the visual analysis of students' predicted knowledge status:

(1) After practicing the recommended 12 educational videos, the students' mastery of the 7 knowledge points involved in them increased significantly, which demonstrated the significance of the educational video resource recommendation model proposed in this paper.

(2) After the first recommendation of educational video No. 300, the students' mastery of knowledge point No. 45 was relatively high, and thus no such educational video was recommended again in the subsequent short period of time. And after recommending educational video No. 303, students' mastery of knowledge point No. 9 remained low, so educational video No. 632 related to knowledge point No. 9 was subsequently recommended. This reflects that the model can compare students' mastery of different educational videos and thus make effective recommendations.

(3) The intelligent body in the educational video resource recommendation model observes that the student's mastery of knowledge point No. 41 is relatively weak, so it recommends the educational video No. 294 that is related to it. After realizing that the student has not mastered the content of the educational video, the intelligent body successively recommends educational videos No. 295 and No. 296 to improve the mastery of knowledge point No. 41. This suggests that the model will make targeted recommendations for educational videos that are more difficult to master.

(4) When students study the recommended educational video, the mastery of the knowledge points not covered in that educational video also decreases slowly over time, which is in line with the law of Ebbinghaus' forgetting curve. This phenomenon indicates that the LSTM-based knowledge tracking model proposed in this paper can achieve the theoretical effect of modeling the forgetting behavior of students in the process of learning, and thus better modeling the knowledge state of students.

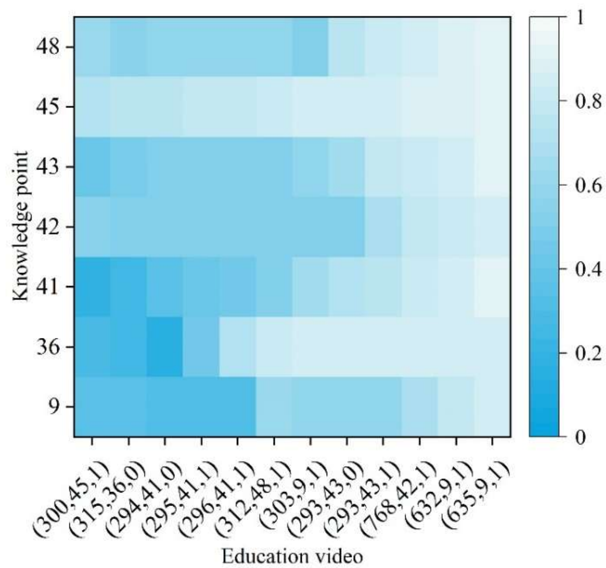



Figure 6: Visual Analysis of Predicted knowledge states

## IV. Artificial intelligence-based personalized learning content push system design

Based on the proposed personalized recommendation model of educational video resources, this chapter completes the design of the automated push system for students' personalized learning content with the support of artificial intelligence.

### *IV. A. System technical architecture*

The system adopts both C/S architecture and B/S architecture. Among the three types of user terminals, PC is B/S architecture and accesses the system through browser, while cell phone and smart watch are C/S architecture and access the system through client. Each device communicates wirelessly with the server based on the TCP/IP protocol, and the server sets up the communication interface, and the corresponding sub-functional interfaces are set up under the communication interface, such as data query, file transfer and other interface functions. The server interface is always in the listening state, when listening to the browser or the client's specific instructions, the corresponding interface triggers a function to return the web page data to the browser to display or a certain format of data to the client to realize the remote function call. Among them, a variety of user-side requests and server-side return data are standardized, formatted language.

Considering that the system has more functions and uses machine learning algorithms, it is difficult to load the system pressure with a single server architecture, so the system adopts a distributed architecture. The system is divided into different microservices according to functional modules, and different microservices can run on different servers and call each other through pre-designed interfaces to complete the teaching assistance work.

The main technologies used in the system are shown in Figure 7. The server side of the system adopts the development method of front-end and back-end separation, which is deployed on Linux server. The back-end part is developed based on the distributed architecture of SpringCloud framework, and the development language is JAVA, and the front-end part is developed based on Vue framework. The client is based on Android platform, the development language is JAVA, and the database is Mysql.

The system uses Gateway to implement a gateway service that controls the flow of data after a request enters the system. Nacos is used to control the registration, discovery, and configuration of each microservice. OAuth2 is used to manage the privileges of the system, controlling the access rights of teachers and students to different functions. Based on Redis, we implement the distributed locking and caching functions of the distributed system to solve the problem of multiple concurrent requests, global login and fast reading of cached data in the distributed system. Realize distributed global transaction management based on Seata, which controls the transaction execution and rollback process at the business layer and reduces the database pressure. Implement the message mechanism through gexin, which can send message push to the client.

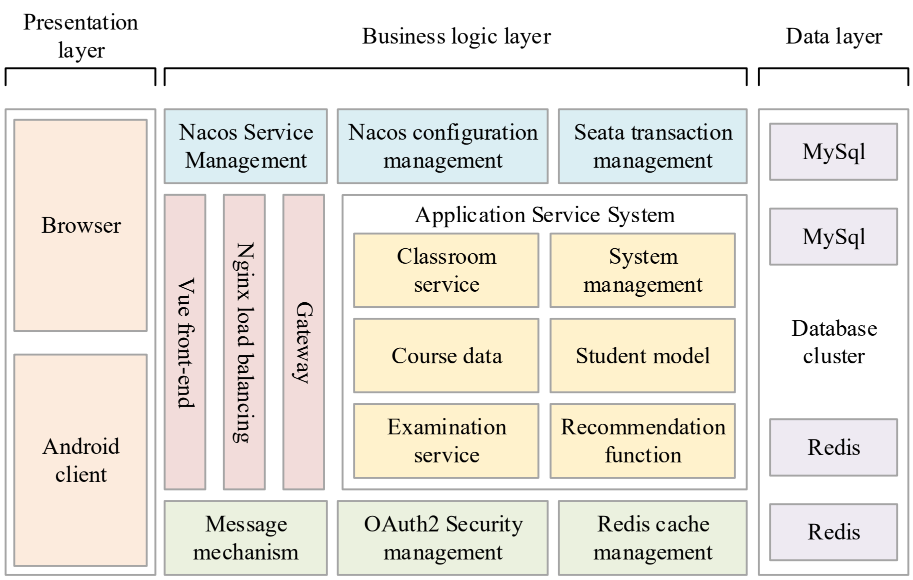



Figure 7: Technical Architecture

### *IV. B. Core business realization*

The main business process of the system is shown in Figure 8, which can be specifically categorized into five application scenarios:

(1) Pre-preparation: Teacher terminal uploads teaching content and knowledge point data. The server receives the uploaded data from the teacher's end and generates the quantitative knowledge point data set. The server receives the uploaded data from the teacher's end and generates the quantitative knowledge point dataset. It collects all the students' past performance data and personality questionnaires to generate the initial student model.

(2) Before each lesson: The server generates a recommended teaching plan based on the student model and knowledge points and pushes the plan to the teacher terminal. The teacher's terminal receives the recommended teaching plan from the server, confirms or modifies it, and then sends the teaching plan to the server. The server receives the teaching plan from the teacher terminal and pushes it to the student terminal. The student terminal receives the lesson plan from the server.

(3) In the classroom: the teacher's terminal initiates classroom timing. The server pushes messages to the teacher's terminal at regular intervals to remind him/her. After the teaching is completed, the teacher initiates classroom polling or video Q&A.

(4) Before the exam: the teacher terminal uploads the exam plan. The server generates predicted student performance and teaching plan according to the student model and test plan, and pushes them to the teacher terminal.

(5) After the exam: the teacher's end inputs the exam results. The server updates the word student model and teaching plan based on the exam results.

In the described process, except for the business part in the classroom and after the exam, all other functions can be used at any moment, such as the view of the recommended plan, the modification of student information and course information, the upload of teaching materials and the modification of the exam plan.

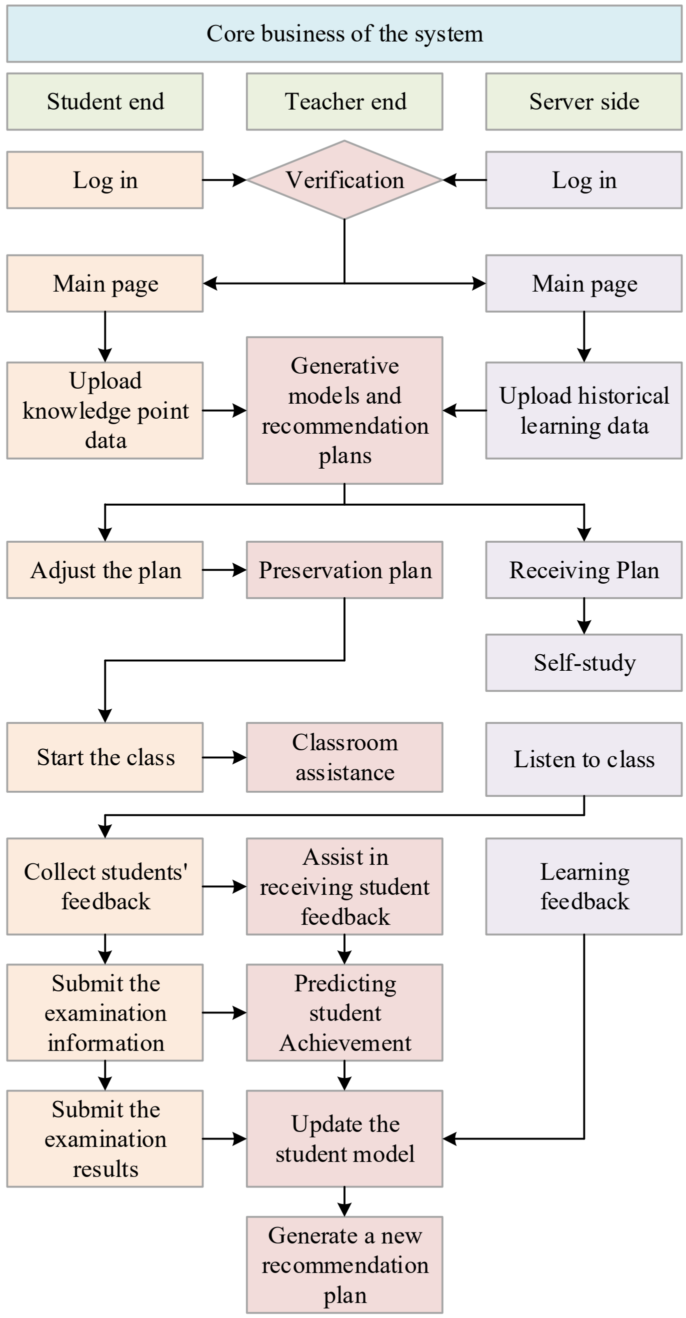



Figure 8: Core business process of the system

## V. Conclusion

The proposed system based on personalized knowledge tracking and educational video recommendation is experimentally verified to have significant application effects. On the POJ dataset, compared with the traditional BKT method, the accuracy of this paper's method is improved by 1.05%, and the AUC value is improved by 2.56%, which verifies the superiority of the model in the knowledge tracking task. On the LLS dataset, this paper's model improves 4.11% in terms of mean square error (MSE) compared to the deep knowledge tracking model DKT. This result indicates that the proposed method is able to accurately track students' knowledge mastery status and recommend appropriate learning content based on students' specific needs.

In addition, the introduction of the personalized exploration strategy enables the system to adjust the recommended learning content according to the students' learning progress and ability, which not only avoids over-recommendation of already mastered content, but also ensures that new content of moderate difficulty can be recommended, improving the efficiency and quality of learning. The parallel matrix computation of the system further improves the computational efficiency of the recommendation algorithm and reduces resource consumption.

The personalized learning content automation push system proposed in this paper can provide students with a more intelligent and personalized learning experience through accurate knowledge tracking and efficient resource recommendation, and has a wide range of application prospects.

## Funding

This work was supported by The Anhui Provincial comprehensive reform and ideological and political ability improvement plan of "three comprehensive education" in colleges and universities: "Online Ideological and Political Education for College Students in the Era of Convergent Media" Counselor Famous Teacher Studio (sztsjh-2023-4-12).

## References

[1] Samala, A. D., Rawas, S., Wang, T., Reed, J. M., Kim, J., Howard, N. J., & Ertz, M. (2025). Unveiling the landscape of generative artificial intelligence in education: a comprehensive taxonomy of applications, challenges, and future prospects. Education and Information Technologies, 30(3), 3239-3278.

[2] Karan, B., & Angadi, G. R. (2025). Artificial intelligence integration into school education: A review of Indian and foreign perspectives. Millennial Asia, 16(1), 173-199.

[3] Brooker, A., Brooker, S., & Lawrence, J. (2017). First year students' perceptions of their difficulties. Student Success, 8(1), 49-62.

[4] Wang, F. (2023). IoT for smart English education: AI-based personalised learning resource recommendation algorithm. International Journal of Computer Applications in Technology, 71(3), 200-207.

[5] Arun Kumar, U., Mahendran, G., & Gobhinath, S. (2022). A review on artificial intelligence based E-learning system. Pervasive Computing and Social Networking: Proceedings of ICPCSN 2022, 659-671.

[6] Vorobyeva, K. I., Belous, S., Savchenko, N. V., Smirnova, L. M., Nikitina, S. A., & Zhdanov, S. P. (2025). Personalized learning through AI: Pedagogical approaches and critical insights. Contemporary Educational Technology, 17(2), ep574.

[7] Gm, D., Goudar, R. H., Kulkarni, A. A., Rathod, V. N., & Hukkeri, G. S. (2024). A digital recommendation system for personalized learning to enhance online education: A review. IEEE Access, 12, 34019-34041.

[8] Yu, H., Wang, J., Murugesan, M., & Rahman, A. S. (2022). Artificial intelligence-based quality management and detection system for personalized learning. Journal of Interconnection Networks, 22(Supp02), 2143004.

[9] Rekha, K., Gopal, K., Satheeskumar, D., Anand, U. A., Doss, D. S. S., & Elayaperumal, S. (2024, May). Ai-Powered Personalized Learning System Design: Student Engagement and Performance Tracking System. In 2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) (pp. 1125-1130). IEEE.

[10] Merino-Campos, C. (2025). The Impact of Artificial Intelligence on Personalized Learning in Higher Education: A Systematic Review. Trends in Higher Education, 4(2), 17.

[11] Yu, H., Miao, C., Leung, C., & White, T. J. (2017). Towards AI-powered personalization in MOOC learning. npj Science of Learning, 2(1), 15.

[12] Murtaza, M., Ahmed, Y., Shamsi, J. A., Sherwani, F., & Usman, M. (2022). AI-based personalized e-learning systems: Issues, challenges, and solutions. IEEE access, 10, 81323-81342.

[13] Singh, A. (2024). The future of learning: AI-driven personalized education. Available at SSRN 5076438.

[14] Akramovna, X. D. (2024). THE ROLE OF ARTIFICIAL INTELLIGENCE IN PERSONALIZED LEARNING. Hamkor konferensiyalar, 1(6), 144-147.

[15] Shireesha, M., & Jeevan, J. (2024). The Role of Artificial Intelligence in Personalized Learning: A Pathway to Inclusive Education. Library of Progress-Library Science, Information Technology & Computer, 44(3).

[16] Li, H., Li, H., Zhang, S., Zhong, Z., & Cheng, J. (2019). Intelligent learning system based on personalized recommendation technology. Neural Computing and Applications, 31, 4455-4462.

[17] Mhagama, J. T., & Garg, K. (2025). A Systematic Review of Educational Recommender Systems: Techniques, Target Users, and Emerging Trends in Personalized Learning. International Journal of Technology in Education Science, 2(1), 79-98.

[18] Neil Andika, Piter Wongso, Faizal Immaddudin Wira Rohmat,Siska Wulandari, Ammar Fadhil, Riswanto Rosi & Nabila Siti Burnama. (2025). Machine learning-based hydrograph modeling with LSTM: A case study in the Jatigede Reservoir Catchment, Indonesia. Results in Earth Sciences,3,100090-100090.

[19] Zhao Yafei & Yang Long. (2024). Constrained contextual bandit algorithm for limited-budget recommendation system. Engineering Applications of Artificial Intelligence,128.

[20] Feng Xu, Kang Chen, Maosheng Zhong, Lei Liu, Huizhu Liu, Xianzeng Luo & Lang Zheng. (2024). DKVMN&MRI: A new deep knowledge tracing model based on DKVMN incorporating multi-relational information. PloS one,19(10),e0312022.