Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

A Study on Designing Multi-Level Computing Models to Enhance the Effectiveness of Computer Education by Combining Generative AI Techniques

Mingxing Zhu[1,✉], Xin Guo[1]

[1] *Zhixing College, Hubei University, Wuhan, Hubei, 430011, China*

ABSTRACT

In today's era, the transformative power of computing is highlighted, and computational thinking has become the core literacy and essential ability of learners, while computer education is an effective carrier for cultivating computational thinking. The article firstly researches the theory related to collaborative filtering and generative adversarial recommender system. Then it combines SeqGAN with traditional CF algorithms, proposes to use sequence generative adversarial network for missing data prediction, and makes appropriate improvements to SeqGAN to make it suitable for generating scoring data, and then further designs a computer teaching system based on this model. The article launches performance testing experiments on Ali's real dataset UserBehavior, and conducts experiments on the effect of computer education with the students of computer application major in a secondary school as the research object. The results of the study show that in the comparative analysis of the pre-test and post-test of computational thinking of the experimental class, the mean of the total score of computational thinking of the experimental class in the pre-test and post-test is 71.17 and 78.35, respectively, and the post-test is more than 7 points higher than the pre-test. It can be concluded that the teaching model of multilevel computational modeling designed in this paper promotes the development of students' computational thinking and academic performance, improves students' learning attitudes, and increases classroom participation.

Keywords: generative adversarial networks, SeqGAN, CF algorithm, multilevel computational model, computer education

1. Introduction

The rapid development of computer technology is changing our lifestyle and career needs. In this digital age, computer education is not only a basic requirement, but also becomes the key to career and future employment [1-2]. How to improve the quality and effectiveness of computer education in

✉ Corresponding author.
E-mail address: mxingz@163.com (M. Zhu).
Received 05 March 2024; Revised 20 June 2024; Accepted 15 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-216
© 2025 The Author(s). Published by Combinatorial Press. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

colleges and universities has become a pressing issue.

Artificial intelligence is no longer a future science fiction, it has become a powerful assistant in our education field [3]. Through personalized learning, intelligent teaching and data analysis, AI can provide students with a more challenging and specifically tailored learning experience [4-5]. As the global technological revolution continues to evolve, computer technology has profoundly changed the way we live and work. As educators in the new era, we need to keep innovating to meet the needs of students and provide them with the best educational experience [6-7]. Especially for computer education in colleges and universities, it has become an urgent task to better train students with computer skills [8].

Artificial intelligence is no longer just a concept in science fiction or movies; it has been deeply integrated into our daily life, especially in the field of education. Through personalized learning, data analytics, and intelligent teaching, AI can provide students with a more challenging and highly customized learning experience, thus increasing their competitiveness in the computing field [9-11]. In the traditional education model, students are usually forced to learn at the same pace and in the same way. However, each student learns in a unique way and at a unique pace. Artificial intelligence can provide personalized learning paths by analyzing students' learning styles, needs, and progress [12-13]. This means that students who grasp knowledge quickly can move forward faster, while students who are confused about certain concepts can receive additional support. This customized approach helps students to better understand and master computer skills while increasing their self-confidence [14-15].

Artificial intelligence can also help educators better understand students' needs and performance through data analytics, and by monitoring students' academic performance and study habits, AI can provide educators with valuable insights that can help them adjust their teaching methods to meet students' needs [16-17]. Such accurate predictions can help reduce students' academic frustration and improve their academic achievement. Artificial intelligence technologies are not meant to replace teachers, but to give them new roles. Teachers can act as tutors and mentors of learning, guiding students to explore the world of computer science more deeply [18-20]. Teachers can also utilize AI data to better understand their students' needs and provide them with more targeted support. This intelligent approach to teaching and learning will make education more interactive and engaging, helping to stimulate student interest and creativity [21-22].

In the broad arena of computer education, the integration of intelligent educational technologies not only crosses the boundaries of traditional teaching, but also tailors a more personalized, efficient and interactive learning experience for students, which greatly accelerates the pace of education modernization [23-24]. By applying AI technology more widely, we can provide better education and produce more students with computer skills, laying a solid foundation for their careers. However, we also need to recognize the limitations of AI technology and continue to explore and innovate in order to realize its full potential [25-27]. Only in this way can we better meet the challenges of the digital age and pave the way for our students' future.

Literature [28] systematically reviewed the research literature related to the game education model in computer education practice, pointing out that these research methods have shortcomings such as high subjectivity and small samples, and arguing that a more scientifically rigorous approach is needed for assessment and research. Literature [29] searched journal articles on computer education and found that the research themes developed as popularization of computer education, independent multimedia learning, network computer collaborative learning and online and learning in the digital age, which provided a reference to the development of the theme vein for the research on the related themes. Literature [30] conducted a pedagogical experiment to evaluate the performance of mobiledu in the computer education classroom, and found that mobiledu stimulated students' computer learning potential and corrected students' interest in computer system design learning. Literature [31] describes the Lab4CE computer distance learning laboratory that provides a practical platform for

students' computer learning and positively contributes to students' computer performance. Literature [32] summarizes the research on educational technology, which mainly contains computer-assisted instruction based on communication technology (ICT) and computer technology-based (CAI) assisted instruction, and in general, educational technology has played a positive role to a certain extent. Literature [33] developed an educational framework adapted to higher degree programs with computing and provided an international tool to describe the learning of computer competencies, which helps to build a good learning environment for computer courses. The above studies have explored the current status and development of computer education in depth from the dimensions of computer education development, computer teaching tools, educational technology empowering computer teaching and learning as well as the design of teaching and learning frameworks, but there are fewer studies that deal with intelligent education.

On the other hand, AI technology is commonly used in education, and the research around AI technology in education practice is mainly investigative, with the main goal of exploring the value, advantages and shortcomings of AI technology in education. Literature [34], in a study on the application of OpenAI ChatGPT's language generation model in the field of education, learned that the majority of users have a positive attitude towards AI technology in education, and pointed out that it is possible to mitigate the bad usage of the technology based on clear policies and guidelines, and to guide it to be used in education in a scientific, rational and legal way. Literature [35] combined with library literature research, pointed out that AI technology has been commonly used in education, the main forms of expression are, virtual tutors, voice teaching assistants, teaching translation and personalized learning, etc., and believed that the emergence of AI technology has effectively alleviated and shared the heavy learning tasks and pressure of teachers. Literature [36] is based on empirical investigation methods to discover the deficiencies of AI in educational practice, and to improve the framework of the use of AI technology in educational applications around the three aspects of teaching, governance and practical operation. Literature [37] describes the positive role played by AI technology in education, including assisting teaching and broadening teaching content and thinking, and points out that AI technology will replace non-teaching positions to a certain extent. Literature [38] delves into the value potential of the introduction of AI technology into education, basing its in-depth analysis on existing research on educational models of AI technology, and provides theoretical references for understanding how AI technology can reshape education.

The article first introduces the gaming idea and realization principle based on collaborative filtering and generative adversarial network GAN. Then it combines SeqGAN with traditional CF algorithms to use SeqGAN for missing data prediction, which predicts more realistic learner behavioral preference data based on the time-series relationship of users' historical behavioral data, so as to fill in the scoring data. After completing the model establishment, the system architecture as well as the system functions are determined, and the system application design and implementation are completed according to the requirements, and a series of model performance checking experiments are carried out on the real dataset UserBehavior. Finally, an empirical study is carried out with the students of computer application major in a secondary school in a province as the research object, and relevant data are collected in teaching practice to verify the application effect of the computer model proposed in this paper in computer education in terms of computational thinking measurements, academic achievement tests, problem exploration records, classroom observation records, and interviews and analyses.

2. Computerized teaching system based on generative AI multilevel computational modeling

2.1. Collaborative Filtering and Generative Adversarial Networks

2.1.1. Collaborative Filtering Recommendation Algorithms:

1) Content-based recommendation. The core idea of content-based recommendation strategy: based on the characteristics of the user's historically preferred item content, recommend items similar to the user's preferred items. Two key issues are as follows: first, how to represent the user's preference? Second, how to represent the characteristics of the items to be predicted? First the user's preference is characterized by the user's implicit feedback or explicit feedback. Then the content features of the items to be predicted are computed. Finally, the features of user preferences and the features of the items to be predicted are matched for similarity, and after matching, they are sorted for recommendation.

2) Collaborative Filtering Recommendation. The core idea of collaborative filtering: given a target user A, based on the ratings of A on the project and the historical ratings of all users except A on the project, find the set of users B with higher similarity to A. Because user A may like the same project as set B, recommend the project that user A likes to B. Assuming that there are m user and i items, a co-scoring matrix of $m*n$ is formed M. Where M_{ij} represents the scoring of item j by user i, and if M_{ij} is null (missing value), it means that user i has not scored item j [39].

A. Memory-Based Collaborative Filtering

The most commonly used in memory-based collaborative filtering is the nearest neighbor approach. The main process of the nearest neighbor technique: first, a number of nearest neighbor users of the target user are obtained based on the similarity of user ratings. Then, the ratings of the predicted items are weighted and counted based on the ratings of the nearest neighbor users. Finally, the final ratings of the predicted items are obtained and ranked for recommendation. Memory-based collaborative filtering includes user-based collaborative filtering and item-based collaborative filtering. In both user-based and item-based collaborative filtering, calculating the similarity occupies a very important position. The commonly used methods for calculating similarity are as follows: euclidean distance, cosine similarity, Pearson's correlation coefficient and Jaccard's similarity coefficient.

Euclidean distance. It is an alias of Euclidean distance and refers to the straight line distance between two points in space. It can be used to calculate the absolute distance between user i's rating vector u_i and user j's rating vector u_j in a n-dimensional vector space. The Euclidean distance $d(u_i,u_j)$ is shown in equation (1).

$$\mathrm{d}(u_i,u_j)=\sqrt{\sum(\vec{x}_i-\vec{x}_j)^2}\in[0,\infty) \tag{1}$$

The normalized Euclidean distance, as shown in equation (2) below.

$$\sin(u_iu_j)=\frac{1}{1+d(u_i,u_j)}\in(0,1] \tag{2}$$

Pearson's correlation coefficient. It is mainly used to measure the degree of linear correlation between variable Y and variable X. The range of this coefficient is [-1, 1]. Less than 0 represents negative correlation, more than 0 represents positive correlation, and equal to 0 is not correlated. The specific formula is shown in equation (3).

$$\sin(i,j)=\frac{\sum_{x\in I_{ij}}(R_{i,x}-\bar{R}_I)(R_{j,x}-\bar{R}_J)}{\sqrt{\sum_{x\in I_{ij}}(R_{i,x}-\bar{R}_I)^2\sum_{x\in I_{ij}}(R_{j,x}-\bar{R}_J)^2}} \tag{3}$$

Cosine Similarity. Cosine similarity, which also becomes cosine similarity, is evaluated between two vectors by calculating the cosine of the angle between them. The exact formula is shown in equation (4).

$$\sin(x,y)=\frac{\sum_{i=1}^{n} R_{x,i}R_{y,i}}{\sqrt{\sum_{i=1}^{n} R_{x,i}^2}\sqrt{\sum_{i=1}^{n} R_{y,i}^2}} \tag{4}$$

Jaccard similarity coefficient. The ratio of the number of elements of the intersection of two sets C (the set of items rated by user i) and D (the set of items rated by user j) in the concatenation of C and D is known as the Jaccard coefficient of the two sets, and the Jaccard similarity coefficient is a measure of similarity between the two sets. The specific formula is shown in equation (5).

$$\sin(i,j)=\frac{|S(i)\cap S(j)|}{|S(i)\cup S(j)|} \tag{5}$$

where $S(i)$ denotes the set of items rated by user i and $S(j)$ denotes the set of items rated by user j.

B. Model-based collaborative filtering

Model-based collaborative filtering mainly uses algorithms such as machine learning or data mining to build a model, train the model using data from users and items, and use the trained model to predict the ratings of items by target users. The main models include Bayesian model, matrix decomposition, clustering model and so on.

(1) Matrix decomposition. Among model-based collaborative filtering, matrix decomposition is the most popular. Suppose there are m users and n items. The matrix decomposition can be used to learn the potential vector M of m users and the potential vector N of n items to obtain the rating matrix $R(R=M*N^T)$. The principle of matrix decomposition is shown in Fig. 1, which maps the original rating matrix R into the inner product of the transpose of two matrices M and N of dimension f in the hidden space. Each user u can be represented using vector $m_u \in R^f$ and each item i can be represented using vector $n_i \in R^f$. For the target user u, to predict the rating $\hat{r}_{ui}$ of item i by user u, it is shown in equation (6).

$$\hat{r}_{ui}=m_u n_i^T \tag{6}$$

From Eq. (6), it is key to get the vectors for each user and item. The user's predicted rating for the item can be obtained by calculating the inner product of the user vector and the item vector. Regularization is often done to prevent overfitting from arising. The specific formula for the matrix decomposition loss function is shown in (7).

$$L=\sum_{u,i\in R}(r_{ui}-\hat{r}_{ui})^2+\lambda(\|m_u\|^2+\|n_i\|^2) \tag{7}$$

In equation (7), r_{ui} is the true rating of item i by user u, $\hat{r}_{ui}$ is the predicted rating of item i by user u, λ is the regularization penalty coefficient, and $\|\cdot\|^2$ is the second paradigm.

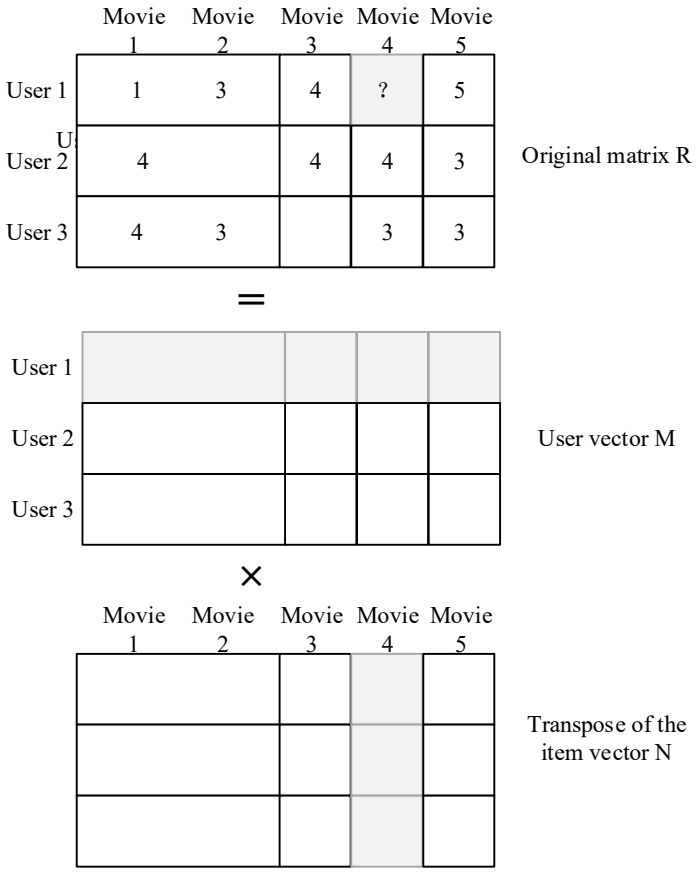

Fig. 1. Schematic diagram of matrix decomposition

(2) Bayesian Modeling. A Bayesian model is represented using a directed acyclic graph (D, B, Θ). D represents the set of vertices of an acyclic graph. B represents the set of edges of an acyclic graph, and an edge can connect two vertices. Θ then represents a table of conditional probabilities, where each element of the table represents the probability that two vertices are connected. The conditional probability is the probability that event A occurs when random event B occurs $P(A \mid B)$. Bayes' theorem, on the other hand, is a theorem about the conditional probability of random event A and random event B, as shown in equation (8).

$$P(A \mid B) = \frac{P(AB)}{P(B)} \tag{8}$$

(3) Clustering models. The clustering model performs clustering based on similarity, which can be measured using the Perkovsky running distance. For example, given two data points, $X = (x_1, x_2, x_3, ..., x_n)$ and $Y = (y_1, y_2, y_3, ..., y_n)$, the Minkowski distances for X and Y are shown in equation (9).

$$d(X, Y) = \sqrt[q]{\sum_{i=1}^{n} | x_i - y_i |^q} \tag{9}$$

2.1.2. Generating an Adversarial Network GAN. GAN is a generative adversarial network that estimates generative models through an adversarial process.The framework structure of GAN is illustrated in Fig. 2. As seen in the figure, GAN trains two models simultaneously. The generator G is used to capture the probability distribution of the data and the discriminator D is used to discriminate whether the samples come from the training data or the data generated by the generator, where G and

D are ordinary deep neural network models [40]. GAN is based on the very large and very small game rather than optimization problem, which has a value function as shown in Eq. (10).

$$min_G maxV(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{10}$$

where $p_z(z)$ is an a priori noise and $D(x)$ denotes the probability that x is from the real data and not from G. The training process is as follows. Train D to maximize the correct recognition of the real samples as positive samples and the recognition of the generated samples as negative samples, and train G to minimize $\log(1-D(G(z)))$. The process of adversarial training is as follows, first train D, take a batch of real samples and a batch of hidden variables, the hidden variables are used as inputs to the generation of G to get the generated samples, and at this time, keep G's network parameters unchanged, and update D's network parameters. After that train G, also take a batch of hidden variables through G to get the generated samples, at this time keep the network parameters of D unchanged and update the network parameters of G.

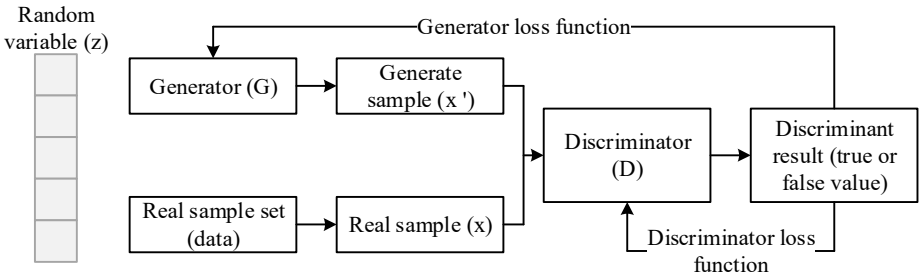

Fig. 2. Structure of the generated adversarial network

2.2. Collaborative Filtering Recommendation Model Based on SeqGAN

2.2.1. Model structure. The model first processes the dataset into a labeled score vector dataset with time series according to the data processing method. The model structure is shown in Fig. 3. The Generator part of the figure is the generator model of SeqGAN, in this paper, LSTM is chosen as the generator of the model, the generator generates the fake label rating sequence $Y = y_1, y_2, \cdots, y_t$ by random noise z, where y_t represents the rating vector of the user at the moment t. The figure shows the specific training adversarial process of the model, which passes the fake label scoring sequence Y generated by the generator and the real user label vector sequence $X = x_1, x_2, \cdots, x_t$ into the discriminator, and the discriminator outputs the probability values of the real scoring sequence X and Y respectively, p^+ if the real scoring sequence X is identified as the real scoring sequence, and p^- if the scoring sequence Y generated by the generator is identified as the real scoring sequence. The probability value p^- is then used as a reward to guide the generator to send generation updates, prompting the generator to generate more realistic user-rated sequences of labels as much as possible. The discriminator ends training and saves the model when it cannot distinguish whether the rating sequence Y generated by the generator is a generated sequence or a real rating sequence, i.e., $p^- = 0.5$.

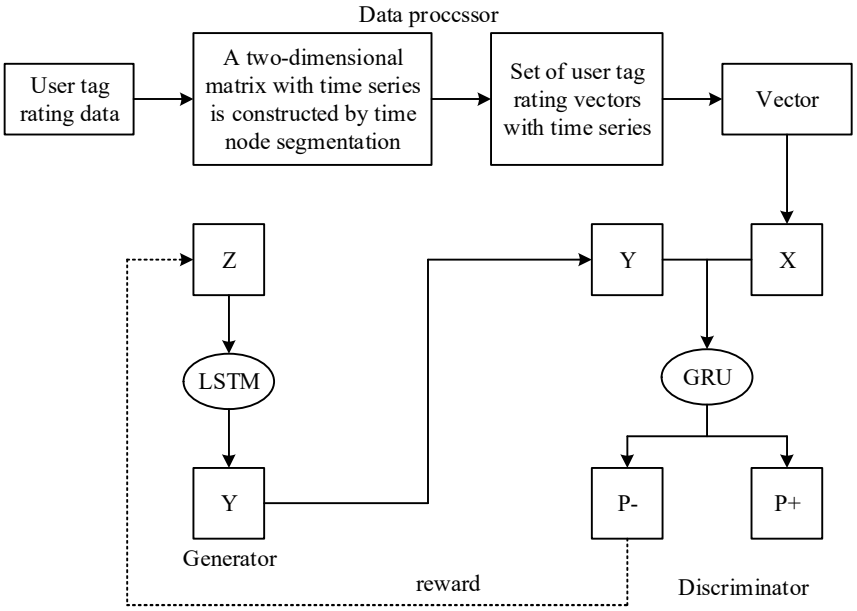

Fig. 3. Model structure

2.2.2. Model Generator. The long and short-term memory network LSTM has a better effect on temporal data processing, and it can use the user's historical labeled rating data to uncover the user's behavioral preference, so as to predict the new labeled rating data, so this paper adopts the long and short-term memory network as the generator of SeqGAN G_θ (θ is the parameter) [41]. leakyrelu function not only accelerates the convergence speed, and solves the gradient vanishing problem, it can also provide a good control of the information written into the input gate, so that the user's temporal static preference and dynamic preference can be processed at the same time, so in this paper, we use the leakyrelu activation function in the input gate (i_t) of the long and short-term memory network.

In order to avoid the generator's overdependence on the original data and the problem of model overfitting, random noise z is used to replace the real rating sequence input to the generator, and this method enables the generator to predict more realistic user preference data.

As shown in Eq. The LSTM begins its training by first generating an initial hidden state h_1^g through random noise z. Where y_t is a sequence of generated scoring vectors, the specific generation process is shown in Eq. Generator $G_\theta = \{y_1, y_2, \cdots, y_T\}$.

$$i_t = RELU(W_i \cdot \left[h_{t-1}^g, y_t \right] + b_i) \tag{11}$$

$$h_1^g = LSTM(z, h_0^g) \tag{12}$$

$$h_t^g = LSTM(y_t, h_{t-1}^g) \tag{13}$$

$$y_t = W \cdot h_t^g + b \tag{14}$$

2.2.3. Model discriminator. The gated recurrent unit GRU has achieved better results in text categorization tasks, so in this paper, GRU is chosen as the discriminator D_ϕ (ϕ is a parameter) to assist the training of the generator G_θ. $D_\phi(Y_{1:T})$ represents the probability that the generated rating sequence $Y_{1:T} = \{y_1, y_2, \cdots, y_T\}$ may be a true rating sequence. The rating sequence generated by generator G_θ is used as a negative example and the real labeled rating sequence is used as a positive example to train the discriminator model D_ϕ for discrimination.

As shown in Eq. The discriminator ultimately uses a sigmoid function to evaluate the probability that each node in the generated scoring sequence may be a real scoring data, and accumulates the value of this estimated probability, and finally takes its average value as the probability that the sequence may be a real scoring sequence, as shown in Eq.

$$h_t^d = GRU(y_t, h_{t-1}^d) \tag{15}$$

$$p(y_1, y_2, \cdots, y_{t-1}) = \sigma(Wh_t^d + b) \tag{16}$$

$$D_\phi(Y_{1:T}) = \frac{1}{T}\sum_{i=1}^{t} p(y_i) \tag{17}$$

2.2.4. Strategy Gradient. In order to better assist the model generation against training, the learning process of the action value function $Q_{D_0}^{G_\theta}(s,a)$ is carried out through a reinforcement learning algorithm with the expression shown in Eq. (18), where $Q_{D_0}^{C_\theta}(s,a)$ denotes the expected reward that can be obtained after taking the action a starting from the state s. Using the discriminator D_ϕ as the reward function, the estimated probability value of D_ϕ is regarded as the actual reward obtained and dynamically updated iteratively D_ϕ while guiding the training of the generator for further optimization of the generator.

$$Q_{D_\varnothing}^{G_\theta}(a = y_T, s = Y_{1:T-1}) = D_\varnothing(Y_{1:T}) \tag{18}$$

Once the obtained scoring sequences are more realistic, the model will be re-trained on the discriminator with the model description shown in Equation (19).

$$\min_\varnothing(-E_{X\sim p_{data}}[\lg D_\varnothing(X)] - E_{Y\sim G_\theta}[\lg(1 - D_\varnothing(Y))]) \tag{19}$$

The objective of the generator G_θ is to generate a sequence of label scores based on the starting state s_0 and maximize its expected final reward as much as possible, with the objective function $J(\theta)$ shown in equation (20).

$$J(\theta) = E[R_T \mid s_0, \theta] = \sum_{y_1 \hat{\mathrm{o}} y} G_\theta(y_1 \mid s_0) \cdot Q_{D_\varnothing}^{G_\theta}(s_0, y_1) \tag{20}$$

where R_T denotes the reward after generating the complete sequence of ratings.

During adversarial training of the model, the generator is updated each time a new discriminator is obtained. The optimization approach of the model is based on the strategy gradient, which relies on maximizing the long-term reward. Then the gradient is solved for the objective function $J(\theta)$ as shown in equation (21).

$$\nabla_\theta J(\theta) = \sum_{t=1}^{T} E_{Y_{1t-1}\sim G_\theta}\left[\sum_{y_t \hat{\mathrm{o}} y} \nabla_\theta G_\theta(y_t \mid Y_{1:t-1}) \cdot Q_{D_0}^{G_\theta}(Y_{1:t-1}, y_t)\right] \tag{21}$$

where y_t is the sequence of ratings generated by G_θ. The expectation E can be approximated by sampling and then updating the parameters of the generator as shown in Eq. (22).

$$\theta \leftarrow \theta + \alpha_{h\theta}^{\nabla} J(\theta) \tag{22}$$

In the above equation, the optimization algorithm with adaptive learning rate such as Adam and RMSprop is used, α_h denotes the learning rate in step h, and $\alpha_h \in R_+$.

2.3. Computerized teaching system design

2.3.1. System architecture. This system is a B/S framework, mainly using the MVC model to decouple the system modules.B/S is also known as the web page display mode, also known as the browser/server, because the use of the browser can be directly carried out in the system's knowledge of the recommendation, the server is working in the back-end. This structure accesses the network through the browser and then communicates directly with the user. The main features are: simple operation, no additional consideration of the application, you only need to enter the access address in the browser, and then you can log in to access, high maintainability, the client as well as the system does not need to make changes, only need to be upgraded at the server.

B/S structure is the integration of the browser, server and database, so it is a kind of network architecture, which divides the work in business processing, and provides support for system operation [42]. B/S structure is also based on the development of the C/S structure, which complements and improves the former, and improves the efficiency of the work of the client and the server, and also improves the quality of the system, so it has good applications. B/S structure is also developed based on C/S structure, which complements the former and improves the quality of the system while improving the efficiency of the clients and servers.

MVC is an acronym for Model, View and Controller. Typically, we can divide the J2EE web application structure into three independent and related layers, namely the model, view and controller, each of which accomplishes a relatively fixed function in the application. The model layer is the functional layer used for data processing in the J2EE web application structure, and the main operational functions involve database access and control. The view layer is used in the J2EE web application structure to present the data obtained from the model layer in HTML, CSS and JavaScript. Control layer is a J2EE WEB application used to handle the user's business logic page flow and permission control and other functions, mainly to achieve the function of the system to respond to the user's request for operation, the control process of this business logic, the data used in the database, through the call interface to achieve the call process on the model layer, and then the model layer to return to the data processing in a certain form to display to the user to view and conduct the subsequent operations. It can be seen that the WEB application through the hierarchical division of the architecture, can be better to make the class to maintain independence, and make the different functions of the class call each other more flexible, and at the same time can provide class reuse, so that the general function of the operation of the WEB application code is more easy to extend and reuse.

Based on the system functionality, the system is divided into data layer, algorithm layer, business layer, and interaction layer. The model is actually located in the algorithm layer, which obtains data from the database and processes the data according to the business logic. Views are located in the interaction layer and mainly present data to the user. The controller sits between the interaction layer and the business layer and handles user inputs and interactions.

Data Tier: The main task of the data tier is to store data, using the relational database management system MySQL and the distributed file storage system HDFS. Among them, MySQL has the following advantages: first, performance and efficient service stability. Second, MySQL software is small in memory, easy to install and maintain, and low cost. Third, the MySQL community has active users. HDFS has the following advantages: first, high fault tolerance, data can be saved in multiple copies, once the copy is lost, repair can be automatically realized. Second, it is suitable for big data processing, and it can handle GB, TB and PB level data in terms of data size. Third, the use of streaming data access, write once, read many times, can ensure data consistency. Fourth, it can be built on inexpensive machines with high reliability and scalability, supporting the expansion of more than 10k nodes.

Algorithm layer: collaborative filtering recommendation algorithm based on SeqGAN.

Business layer: the development of the business layer is realized by using relevant software development techniques.

User layer: the user layer mainly consists of two kinds of users, ordinary users and administrators.

2.3.2. System functions. The recommendation system provides users with rich functions, which are mainly divided into four modules: database storage module, index construction module, user query module, and computer teaching knowledge recommendation module.

1) Database storage module: it is responsible for storing the basic data information of the paper as well as the topic information and keyword information obtained through model training, and the data storage module stores the data obtained after the preprocessing operation on the dataset into the database. The database storage module stores the relevant contents in the dataset and provides thesis retrieval support for the index building module.

2) Index construction module: taking the relevant content of the database storage module as input, the computer teaching knowledge stored in the database is indexed and constructed using the Lucene search engine tool.

3) User Retrieval Module: this module mainly provides users with the function of searching papers. In addition, the thesis retrieval module also serves the recommendation module, filtering the thesis according to the user-selected fields, keywords and other information, generating a candidate set of thesis for the user, and the recommendation algorithm finally selects the thesis from the candidate set.

4) Paper Recommendation Module: The paper selected by the user is used as the input text for the recommendation algorithm, and the recommendation model training module mainly uses the model in this paper and the traditional model, and calculates the recommendation list for each user by utilizing the user's history record and the paper's text information. Taking the text summary input by the user as input and the title as output of the text summary, the model is trained to complement the user's rating matrix of the paper, and finally the Top K-rated recommendation candidate list is generated for each user by the Maximum Heap Sorting algorithm. The paper pushing process is shown in Fig. 4.

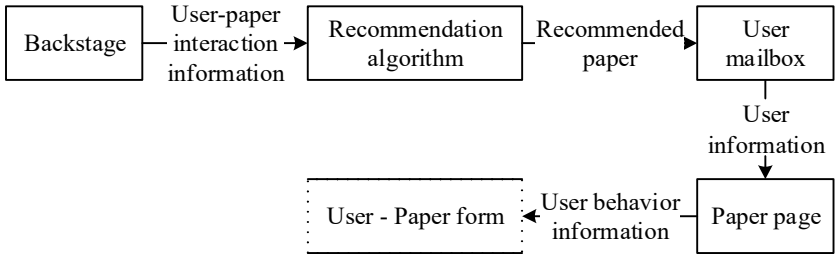

Fig. 4. Paper push process

3. Experiments and analysis of results

3.1. Model performance test

3.1.1. Experimental setup. In this paper, we select the UserBehavior dataset provided by Alibaba, which is organized similarly to MovieLens-20M. By comparing the experiments, this study can better go to discuss whether the GAN-based behavioral path recommendation algorithm can effectively solve the data sparsity problem of the collaborative filtering algorithm, and also more fully discover the advantages and disadvantages of the algorithm in the recommendation evaluation index, which is of great significance to draw experimental conclusions as well as to get the shortcomings in the design. The abbreviations of each algorithm against the display page are shown in Table 1.

Table 1. Each algorithm is a short name for the display page

Numbering	Algorithm	For short

Method 1	Collaborative filtering recommendation algorithm	Item-CF
Method 2	A collaborative filtering recommendation algorithm based on the knowledge map	BR-CF
Method 3	Using the behavior path of the generated data to filter the recommendation algorithm	SeqGAN1
Method 4	Based on GAN's behavioral path collaborative filtering recommendation algorithm	SeqGAN2
Method 5	The hybrid recommendation algorithm for Item-CF and BR-CF	Mix1
Method 6	The hybrid recommendation algorithm for Item-CF and GAN CF	Mix2
Method 7	The hybrid recommendation algorithm for Item-CF and GAN_BR CF	Mix3
Method 8	The cascade recommendation algorithm for Item-CF and BR_CF	Combine
Method 9	The cascade recommendation algorithm for Item-CF and GAN CF	Combine1
Method 10	The cascade recommendation algorithm for Item-CF and GAN_BR_CF	Combine2
Method 11	A layered collaborative filtering recommendation based on Spark	SH-CF
Method 12	A recommendation algorithm based on pheromones	Pheromone-based

3.1.2. Experimental data statistics. The classification of behavioral paths in the system of this paper is shown in Table 2, and 2532 behavioral paths are created based on 20069 behavioral data (from 193 users) in the experiment, and 2622 pseudo-user behavioral paths are generated by using SeqGAN to learn these 2532 behavioral paths. At the same time, according to the definition of behavior path categories, these pseudo-user behavior paths are classified, and finally 1825 "PV" dimensions are obtained, of which 163 are in the "FAV" dimension, 382 are in the "CART" dimension, and 252 are in the "Buy" dimension.

Table 2. The behavior path classification in this article system

Data type	Classification	Quantity/piece
Real data	Behavioral data	20069
	Behavior path	2532
	User	193
Generated pseudo-data	Dimension of “pv”	1825
	Dimension of “fav”	163
	Dimension of “cart”	382
	Dimension of “buy”	252
	total	2622

3.1.3. Analysis of the number of referrals. Due to the existence of data sparsity, i.e., when there is no behavioral path category with lower priority in the data, or when there is not enough user data to form a behavioral path, both Algorithm 1 and Algorithm 2 are unable to make a recommendation in this dimension. Therefore, before evaluating the recommendation using accuracy and coverage, the statistics of the number of people who can be recommended for Methods 1 through 7 are first performed, and the statistics of the number of people who can be recommended are shown in Table 3. In all subsequent graphs and tables d_1 dimensions represent the fav dimension, d_2 dimensions represent the cart dimension, and d_3 dimensions represent the buy dimension. As can be seen from the table, in dimension 1 and dimension 2, method 4 in method 1 to method 4 can recommend the largest number of people, and method 7 in method 5 to method 7 can recommend the largest number of people, which can be seen that whether it is in the recommendation algorithm 2 or in the hybrid recommendation algorithm 5, the inclusion of SeqGAN can reduce the recommendation of the case of the empty, and improve the diversity of the recommendation.

Table 3. The number of people who can recommend the number of people statistics

Algorithm	The number of people recommended in the d_1 dimension	The number of people recommended in the d_2 dimension	The number of people recommended in the d_3 dimension
Method 1	25	86	25
Method 2	20	72	29
Method 3	22	75	19
Method 4	51	95	20
Method 5	42	125	55
Method 6	46	120	50
Method 7	53	131	40

3.1.4. Recommended coverage analysis. The recommended coverage of methods 2-4 is shown in Figure 5, where the area of method 4 covers method 2 and method 3, i.e., the coverage of method 4 is consistently better than that of method 2 and method 3 in three dimensions. In fact, in dimension 1, the coverage of method 2 is 0.15% while that of method 4 is 1.5%, a full order of magnitude improvement. This indicates that Method 4 has better ability to uncover the long tail of items than Method 2. Method 3 is a recommendation after using SeqGAN generated data to cover real data, and the difference between its coverage area and the area of recommendation using real data is not large, which on the one hand indicates that the quality of the generated data is not weaker than that of the real data, and on the other hand, it also indicates that it is not that the quality of the generated data is high enough to cause the GAN-based behavioral path collaborative filtering algorithm to excel in the recommendation coverage rate, but that the use of SeqGAN can effectively alleviate the algorithm performance improvement caused by the data sparsity problem.

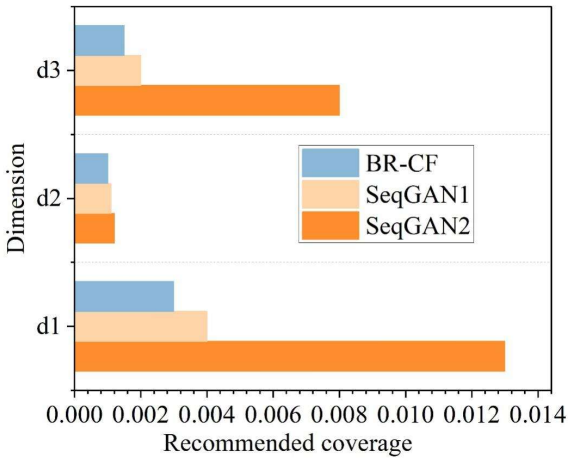

Fig. 5. Method 2-4 recommendation coverage

3.1.5. Comparison of average accuracy:

1) A comparison of the average recommendation accuracy is shown in Fig. 6, where the average accuracy of Method 4 is lower than that of Method 2 and Method 3 in all dimensions, which is completely opposite to Method 4's performance in terms of coverage discussed above, and the study concludes that this is a negative result due to the fact that Method 4 is populated with sparse data. To further validate the above conclusion, this study collects and organizes the shortest distance data computed by Methods 2, 3, and 4 during the recommendation process, and the recommended average shortest distances are shown in Table 4.

Since SeqGAN is the learning of the entire collection of behavioral paths, the behavioral paths it generates are more universal, making the shortest distance calculated by Method 3 in making recommendations smaller compared to Method 2, which uses real data, i.e., the behavioral similarities

are more similar on the whole. And after using SeqGAN to populate the sparse data in the original data, the average shortest distances of each dimension when making recommendations are larger than the average shortest distances of both recommending using only real data and recommending using only SeqGAN-generated data, which, as discussed earlier, is mainly due to the fact that many of the sparse data that could not be recommended originally are populated by the generated data, although they can be recommended, but since the the supplemented data is generalized but not user-specific, resulting in large shortest distances for recommendations. This also coincides to some extent with the conclusion that the recommendation accuracy may decrease after filling sparse data, as a cost of solving data sparsity.

2) Although the average accuracy of method 4 is worse in the figure, it is found that method 7 is optimal in both dimension 1 and dimension 2. Taking the effective number to 3 decimal places, the average accuracy of method 7 is 89.3%, 90.1%, and 73.5% in the three dimensions, while the average accuracy of method 5 is 81.2%, 86.3%, and 72.5% in the three dimensions, and method 7 improves by 8.1%, 3.8%, and 1% in the three dimensions, respectively. In fact, when the recommendation accuracy of 0 is included, the average correctness of method 7 on the 3 dimensions is 76.5%, 79.2% and 27.3%, while at this time the average correctness of method 5 on the dimensions is 55.8%, 66.9% and 32.5%, and method 7 improves over method 5 by 20.7% on dimension 1 and by 12.3% on dimension 2. The results of this study illustrate that the recommendation performance of the hybrid recommendation algorithm is improved by using SeqGAN to alleviate the data sparsity problem, and the hybrid recommendation can also well minimize the negative impact caused by populating sparse data.

3) As shown in the figure, among all the methods, method 7 has the best performance in dimension 1 and dimension 2, and the performance in dimension 3 is relatively not bad, which indicates that the recommendation performance of the hybrid recommendation algorithm of item collaborative filtering and GAN-based behavioral path collaborative filtering proposed in this paper is very excellent, and verifies that the GAN-based behavioral path collaborative filtering recommendation algorithm proposed in this paper is effective. The method proposed in this paper has guiding significance and research value for using GAN to improve the recommendation performance of the recommendation system.

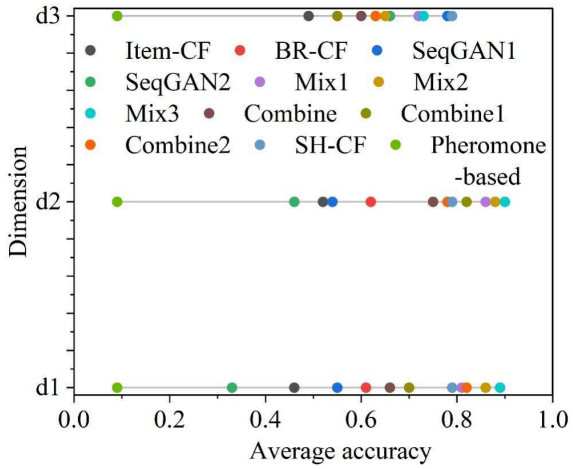

Fig. 6. Comparison of average recommendation accuracy

Table 4. Recommended average shortest distance

Algorithm	The shortest distance of the d_1 dimension	The shortest distance of the d_2 dimension	The shortest distance of the d_3 dimension
Method 2	5.96	9.63	9.3
Method 3	4.75	8.65	8.32

Method 4	9.85	15.62	9.82

4) Although it is important to compare the average accuracy in each dimension, the overall performance of the system can not only look at the performance of each dimension, but also need to compare the performance of the algorithms from a macro point of view. The average accuracy of each dimension is shown in Figure 7. From the content of the figure, it can be seen that the sum of the average accuracy of each dimension, i.e., the comprehensive performance of the algorithms is significantly better than a series of comparison algorithms.The Mix1 method and the Mix2 method are comparable in their comprehensive performance, and the Combine method and the Combine1 method are basically the same in their comprehensive performance, which verifies the validity of the data generated using SeqGAN. At the same time, the Mix3 method performs better than Mix1 and Mix2, and the Combine2 method performs better than Combine and Combine1, which also verifies the effectiveness of using Seq-GAN to generate data to fill the real data, and solving the problem of data sparsity can really improve the comprehensive performance of the algorithm effectively. And comparing all the algorithms, it can be found that the comprehensive performance of Mix3 method is the best among all the algorithms, which further verifies the superiority of the hybrid recommendation algorithm of item collaborative filtering and GAN-based behavioral path collaborative filtering proposed in this paper, and at the same time confirms the effectiveness of the algorithm proposed in this paper.

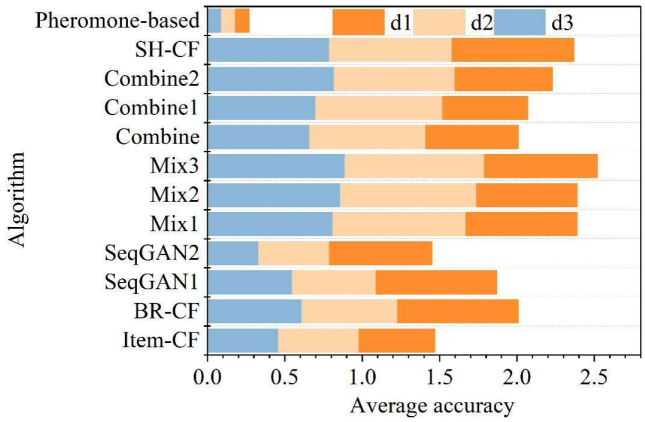

Fig. 7. Average accuracy of each dimension

3.2. Teaching Model Practice for Computational Thinking Cultivation

3.2.1. Preparation for teaching practice:

1) Teaching Practice Program. In order to verify the application effect of the teaching model oriented to the cultivation of computational thinking, the author carries out the teaching practice of the C language course in the computer application major of a secondary school in a province, selects experimental classes and control classes, and mainly examines whether the application of the multilevel computational model designed in this paper in the classroom of computer education can effectively cultivate the computational thinking ability of the students, enhance the students' academic performance, and improve the students' learning attitudes, increase classroom participation. Before starting the teaching practice, two classes of comparable level were identified as the subjects of the study. Computer application sophomore class 1 and class 2, the class type is the same, the students' past performance is basically the same, class 1 has 40 students, class 2 has 40 students, the two classes have the same number of students, with similar male to female ratio structure. In this study, the teaching mode is the independent variable, the experimental class and the control class have different

teaching modes, the first class adopts the systematic assistance of this paper for teaching, and the second class adopts the traditional lecture-practice teaching mode. The dependent variables are students' computational thinking level, theory test scores and skills assessment scores, and the changes in the dependent variables are derived by analyzing the data of questionnaire survey, theory knowledge test, programming skills assessment and problem exploration records.

2) Implementation of the pre-test:

(1) Computational thinking pre-test. Eighty paper questionnaires on computational thinking pre-test were distributed to the first and second computer classes. Independent samples t-test was conducted on the questionnaire pre-test data of the two classes using SPSS 20.0, and the statistics of independent samples t-test for computational thinking pre-test of Class I and Class II are shown in Table 5. As can be seen from the table, the mean values of the five sub-dimensions of computational thinking in Class I and Class II are similar, and the mean values of the total computational thinking scores are also not very different. In terms of Sig. (two-sided) values, the t-test Sig. values of Class I and Class II on the five sub-dimensions of computational thinking are greater than 0.05, and the t-test Sig. value of the two classes on the total score of computational thinking is 0.782, which is also greater than 0.05, which indicates that before carrying out the teaching practice, there is no significant difference between the two classes in computational thinking in general and in each sub-dimension, and that the two classes have the basically the same level of computational thinking.

Table 5. Test the test statistics of the independent sample t before calculating the thought

Dimension	Class	Number	Mean	Standard deviation	t	Sig.(Double side)
Abstraction	Class 1	40	14.84	2.877	-0.149	0.882
	Class 2	40	15.28	2.607		
Decomposition	Class 1	40	13.18	2.726	-0.512	0.624
	Class 2	40	14.18	2.542		
Algorithm	Class 1	40	14.31	2.793	-0.169	0.872
	Class 2	40	14.5	2.662		
Evaluate	Class 1	40	13.78	2.201	-0.302	0.772
	Class 2	40	13.43	2.577		
Recapitulate	Class 1	40	15.05	2.739	0.196	0.851
	Class 2	40	15.08	2.322		
Calculate the total score of thought	Class 1	40	71.24	9.896	-0.263	0.782
	Class 2	40	72.1	8.434		

(2) Programming skills pre-test. The programming skills pre-test was taken as a computerized test, in which students completed the programming operation within the specified time, and the full score was 100 points. Using SPSS 20.0 to carry out independent samples t-test on the programming skills pre-test scores of the two classes, the statistics of independent samples t-test on the theoretical knowledge pre-test of the first class and the second class are shown in Table 6. As can be seen from the table, the mean score of class I is 72.6 and the mean score of class II is 70.62. From the Sig. (two-sided) value, the two-sided Sig. value of the independent sample t-test for Class I and Class II is 0.752, which is greater than 0.05, indicating that the difference between the two classes' scores on the pre-test of programming skills is not significant, and that Class I and Class II have basically the same level of programming skills.

Table 6. The independent sample t test is measured before the theoretical knowledge

Class	Number	Highest	Minimum	Average	Standard deviation	t	Sig.(Double

		score	score	score			side)
Class 1	40	95	49	72.6	10.932	-0.051	0.752
Class 2	40	90	50	70.62	10.692		

3.2.2. Analysis of the effectiveness of teaching practice:

1) Analysis of Computational Thinking Measurement. In order to test whether the teaching mode oriented to the cultivation of computational thinking can improve the level of students' computational thinking, independent samples t-test was conducted on the posttest data of the two classes, and paired samples t-test was conducted on the pre and posttest data of the control class and the experimental class, respectively, which were analyzed as follows:

(1) Comparative analysis of computational thinking posttests of the two classes. Using SPSS 20.0 to conduct independent samples t-test on the computational thinking posttest data of the two classes, the statistics of independent samples t-test on the computational thinking posttest of the experimental class and the control class are shown in Table 7. As can be seen from the table, at the end of the teaching practice, the mean scores of the five sub-dimensions of computational thinking of the experimental class were higher than those of the control class, and the mean of their total computational thinking scores were approximately about 6 points higher than those of the control class. In terms of Sig. (two-sided) values, the t-test Sig. values of the experimental class and the control class on the five sub-dimensions of computational thinking are less than 0.05, and the t-test Sig. value of the two classes on the total score of computational thinking is 0.006, which is less than 0.01, and it produces a significant difference between the two classes in the overall computational thinking scores and in the sub-dimensions. In conclusion, at the end of the teaching practice, the total computational thinking score of the experimental class was higher than that of the control class and showed a difference, and also the scores of the dimensions were higher than that of the control class and showed a difference.

Table 7. Test statistics of independent sample t after calculation

Dimension	Class	Number	Mean	Standard deviation	t	Sig.(Double side)
Abstraction	Laboratory class	40	16.4	1.874	2.211	0.036
	Cross-reference class	40	15.25	2.632		
Decomposition	Laboratory class	40	14.89	2.192	2.105	0.032
	Cross-reference class	40	13.87	2.274		
Algorithm	Laboratory class	40	15.65	2.286	2.088	0.043
	Cross-reference class	40	14.6	2.746		
Evaluate	Laboratory class	40	14.76	1.775	2.185	0.03
	Cross-reference class	40	13.75	2.368		
Recapitulate	Laboratory class	40	16.19	2.055	2.351	0.026
	Cross-reference class	40	15	2.65		
Calculate the total score of thought	Laboratory class	40	78.36	7.491	3.122	0.006
	Cross-reference class	40	72.15	8.208		

(2) Comparative analysis of computational thinking pre- and post-tests in the control class. Using SPSS 20.0, the computational thinking pre- and post-test data of the control class were subjected to paired samples t-test, and the descriptive statistics of the computational thinking pre- and post-test of the control class are shown in Table 8. As can be seen from the table, the mean total score of computational thinking of the control class was 71.5 in the pre-test and 72.13 in the post-test, and the total score of the post-test was slightly higher than that of the pre-test, but the difference was not significant. In

terms of the mean values of the scores of the five sub-dimensions of computational thinking, the scores of the pre-test and the post-test were relatively close to each other.

Table 8. Statistical descriptive statistics were measured before and after the calculation

Dimension	Testing	Number	Mean	Standard deviation	Standard error of mean
Abstraction	Pre-test	40	15.28	2.527	0.485
	Post-test	40	15.04	2.85	0.48
Decomposition	Pre-test	40	14.18	2.37	0.433
	Post-test	40	13.95	2.505	0.435
Algorithm	Pre-test	40	14.5	2.586	0.429
	Post-test	40	14.93	2.26	0.438
Evaluate	Pre-test	40	13.43	2.269	0.423
	Post-test	40	13.39	2.555	0.436
Recapitulate	Pre-test	40	15.08	2.54	0.459
	Post-test	40	15.12	2.521	0.458
Calculate the total score of thought	Pre-test	40	72.1	8.47	1.511
	Post-test	40	72.13	8.263	1.498

The paired sample t-test statistics for the pre and post-test of computational thinking in the control class are shown in Table 9. As can be seen from the table, the two-sided test Sig. value corresponding to the total score of computational thinking in the control class is 0.473, which is greater than 0.05, while the two-sided test Sig. values corresponding to the five subdimensions of computational thinking are all greater than 0.05. It can be concluded that, with the adoption of the traditional teaching mode of C language in the control class, the pre and post-tests of computational thinking are basically the same, and have not produced any significant changes.

Table 9. Test the sample t test statistics before and after the calculation of the thought

Dimension	Pair difference	Number	Mean	Standard deviation	t	Sig.(Double side)
Abstraction	Pre-Post	40	-0.038	1.485	-0.122	0.894
Decomposition	Pre-Post	40	-0.187	1.429	-0.909	0.364
Algorithm	Pre-Post	40	-0.043	0.807	-0.208	0.793
Evaluate	Pre-Post	40	0.02	1.982	0.184	0.865
Recapitulate	Pre-Post	40	-0.062	1.813	-0.167	0.845
Calculate the total score of thought	Pre-Post	40	-0.288	2.127	-0.741	0.473

(3) Comparative analysis of computational thinking pre- and post-tests in the experimental class. A paired-samples t-test was conducted on the computational thinking pre- and post-test data of the experimental class using SPSS 20.0. The descriptive statistics of computational thinking pre- and post-test of the experimental class are shown in Table 10. As can be seen from the table, the mean of the total score of computational thinking of the experimental class was 71.17 in the pre-test, and the mean of the total score reached 78.35 in the post-test, and the post-test was more than 7 points higher than the pre-test. In terms of the mean values of the scores of the five sub-dimensions of computational thinking, the experimental class also scored higher in all the post-tests than in the pre-tests.

Table 10. Calculate the descriptive statistics before and after thinking

Dimension	Testing	Number	Mean	Standard deviation	Standard error of mean
Abstraction	Pre-test	40	14.84	2.658	0.494
	Post-test	40	16.01	1.844	0.344
Decomposition	Pre-test	40	13.18	2.79	0.511
	Post-test	40	15.06	2.022	0.337
Algorithm	Pre-test	40	14.31	2.495	0.441
	Post-test	40	15.82	2.088	0.38
Evaluate	Pre-test	40	13.78	2.3	0.38
	Post-test	40	14.99	2.092	0.391
Recapitulate	Pre-test	40	15.05	2.647	0.494
	Post-test	40	16.63	2.038	0.371
Calculate the total score of thought	Pre-test	40	71.24	9.868	1.778
	Post-test	40	78.35	7.46	1.355

The paired sample t-test statistics for the pre and post-tests of computational thinking in the experimental class are shown in Table 11. From the table, it can be further seen that the means and t-values of the total computational thinking scores and the paired difference scores of each sub-dimension in the experimental class are negative, indicating that the scores of the post-test are higher than those of the pre-test. Meanwhile, the corresponding two-sided test Sig. values are all less than 0.01, indicating that the pre and post test results show significant differences. It can be concluded that the level of computational thinking in the experimental class was significantly improved at the end of the teaching practice oriented to the cultivation of computational thinking.

Table 11. Test the sample t test statistics before and after the calculation of the thought

Dimension	Pair difference	Number	Mean	Standard deviation	t	Sig.(Double side)
Abstraction	Pre-Post	40	-1.509	0.945	-5.261	0.000
Decomposition	Pre-Post	40	-1.728	0.964	-7.62	0.000
Algorithm	Pre-Post	40	-1.335	1.011	-5.93	0.000
Evaluate	Pre-Post	40	-1.423	1.27	-6.661	0.000
Recapitulate	Pre-Post	40	-1.666	1.64	-3.919	0.000
Calculate the total score of thought	Pre-Post	40	-7.293	3.956	-10.028	0.000

2) Analysis of skill test scores. The post-test scores of programming skills in the experimental and control classes are shown in Table 12. As can be seen from the table, 9% of the students in the experimental class scored less than 60 points and 25% of the students in the control class scored less than 60 points, and the proportion of failing students in the control class was much higher than that in the experimental class. Students with more than 70 points in the experimental class accounted for 59%, while students with more than 70 points in the control class accounted for only 36%, and the proportion of students with moderate grades or higher in the experimental class was significantly higher than that in the control class. The percentage of students with more than 90 points in the experimental class is 7%, and the percentage of students with more than 90 points in the control class is 1%, and the percentage of students with high scores in the experimental class is also higher than that in the control class. It can be seen that the experimental class's performance in the post-test of programming skills is better than that of the control class.

Table 12. Experimental class and comparison class programming skills

Class	Below 60	60-69	70-79	80-89	90-100
Class1	9%	25%	37%	22%	7%
Class2	25%	38%	25%	11%	1%

In order to more accurately understand the difference between the programming skills posttest scores of the two classes, the programming skills posttest scores of the two classes were imported into SPSS 20.0, and independent samples t-tests were conducted, and the statistics of independent samples t-tests of programming skills posttests of the experimental class and the control class are shown in Table 13. As can be seen from the table, the mean score of the programming skills posttest of the experimental class is 73.96, and the mean score of the programming skills posttest of the control class is 65.96, and the mean score of the experimental class is higher than that of the control class by more than 6 points. Meanwhile the two-sided Sig. value of the independent samples t-test for the two classes is 0.042, which is less than 0.05, indicating that there is a significant difference between the two classes in programming skills posttest scores.

Table 13. Test the test statistics for the independent sample t after programming skills

Class	Number	Highest score	Minimum score	Average score	Standard deviation	t	Sig.(Double side)
Class1	40	96	56	73.96	11.263	2.265	0.042
Class2	40	91	45	65.95	12.542		

3) Analysis of classroom observation records. A comparison of the average scores of classroom observations of the experimental and control classes is shown in Figure 8. As can be seen from the figure, the average scores of the experimental class are higher than those of the control class in all six observation points. In particular, the gap between the classroom participation scores of the two classes is even greater, and the classroom participation of students in the experimental class is significantly better than that of the control class. It can be seen that the use of the system in this paper to assist in teaching can improve students' classroom participation.

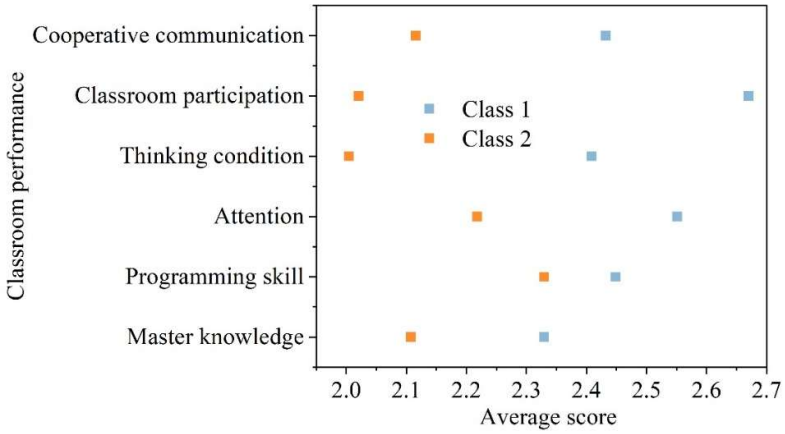

Fig. 8. The experiment class and the comparison class observed the average comparison

4. Conclusion

Based on combing the theories related to collaborative filtering and generative adversarial recommender systems, this study proposes the design of a multi-level computational teaching resource recommendation model combined with generative AI technology, and the application of computerized teaching. The research results are summarized as follows:

The results of a series of experiments on the real dataset UserBehavior show that the average accuracy of Mix3 on three dimensions is 8.1%, 3.8% and 1% higher than that of Mix1 on three dimensions, respectively. The results of this study illustrate that the recommendation performance of the hybrid recommendation algorithm is improved by using SeqGAN to alleviate the data sparsity problem, thus validating the effectiveness of the proposed algorithm in this paper.

In the analysis of programming skills post-test scores of the experimental and control classes. Students with less than 60 points in the experimental and control classes accounted for 9% and 25% respectively, and the proportion of failing students in the control class was much higher than that in the experimental class. It can be concluded that the programming skills post-test scores of the experimental class are better than those of the control class. That is, the use of computer teaching system based on generative AI multilevel computing model can significantly improve the effect of computer education.

Funding

This work was supported by Zhixing College of Hubei University,the 2024 University-level Teaching Reform Research Project —"The Transformation Practice of Computer Education Boosted by Generative AI Technology: A Perspective Based on Large Language Models" (Project Number: XJY202415).

References

[1] Yadav, A. K., & Oyelere, S. S. (2021). Contextualized mobile game-based learning application for computing education. Education and Information Technologies, 26(3), 2539-2562.

[2] García-Peñalvo, F. J., & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. Computers in human behavior, 80, 407-411.

[3] Gorbunova, I. B., & Kameris, A. (2019). Music computer education concept for teachers: Raising the problem. International Journal of Recent Technology and Engineering, 8(2 S4), 913-918.

[4] Denning, P. J., & Tedre, M. (2019). Computational thinking. Mit Press.

[5] Paiva, J. C., Leal, J. P., & Figueira, Á. (2022). Automated assessment in computer science education: A state-of-the-art review. ACM Transactions on Computing Education (TOCE), 22(3), 1-40.

[6] Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. Computers & Education, 126, 296-310.

[7] Passey, D. (2017). Computer science (CS) in the compulsory education curriculum: Implications for future research. Education and Information Technologies, 22, 421-443.

[8] Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. Communications of the ACM, 60(4), 55-62.

[9] Tsai, M. J., Wang, C. Y., & Hsu, P. F. (2019). Developing the computer programming self-efficacy scale for computer literacy education. Journal of Educational Computing Research, 56(8), 1345-1360.

[10] Taslibeyaz, E., Kursun, E., & Karaman, S. (2020). How to Develop Computational Thinking: A Systematic Review of Empirical Studies. Informatics in Education, 19(4), 701-719.

[11] Raja, R., & Nagasubramani, P. C. (2018). Impact of modern technology in education. Journal of Applied and Advanced Research, 3(1), 33-35.

[12] Chiu, T. K. (2024). The impact of Generative AI (GenAI) on practices, policies and research direction in education: A case of ChatGPT and Midjourney. Interactive Learning Environments, 32(10), 6187-6203.

[13] Mata-López, W. A., & Tobón, S. (2018). Analysis of factors associated to the enrollment and demand of computing-related careers. Social Sciences, 8(1), 1.

[14] Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. ACM Transactions on Computing Education (TOCE), 18(1), 1-24.

[15] Chen, X., Zou, D., Cheng, G., & Xie, H. (2020). Detecting latent topics and trends in educational technologies over four decades using structural topic modeling: A retrospective of all volumes of Computers & Education. Computers & Education, 151, 103855.

[16] Lepper, M. R., & Malone, T. W. (2021). Intrinsic motivation and instructional effectiveness in computer-based education. In Aptitude, learning, and instruction (pp. 255-286). Routledge.

[17] Belland, B. R., Walker, A. E., Kim, N. J., & Lefler, M. (2017). Synthesizing results from empirical research on computer-based scaffolding in STEM education: A meta-analysis. Review of Educational Research, 87(2), 309-344.

[18] Ahmad, A., Zeshan, F., Khan, M. S., Marriam, R., Ali, A., & Samreen, A. (2020). The impact of gamification on learning outcomes of computer science majors. ACM Transactions on Computing Education (TOCE), 20(2), 1-25.

[19] Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. ACM Transactions on Computing Education (TOCE), 18(1), 1-25.

[20] Tsarava, K., Leifheit, L., Ninaus, M., Román-González, M., Butz, M. V., Golle, J., ... & Moeller, K. (2019, October). Cognitive correlates of computational thinking: Evaluation of a blended unplugged/plugged-in course. In Proceedings of the 14th workshop in primary and secondary computing education (pp. 1-9).

[21] Warner, J., & Guo, P. J. (2017, August). Hack. edu: Examining how college hackathons are perceived by student attendees and non-attendees. In Proceedings of the 2017 ACM Conference on International Computing Education Research (pp. 254-262).

[22] Wing, J. (2017). Computational thinking's influence on research and education for all. Italian Journal of Educational Technology, 25(2), 7-14.

[23] Touretzky, D., Gardner-McCune, C., Martin, F., & Seehorn, D. (2019, July). Envisioning AI for K-12: What should every child know about AI?. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 9795-9799).

[24] Winkler, R., & Söllner, M. (2018, July). Unleashing the potential of chatbots in education: A state-of-the-art analysis. In Academy of management proceedings (Vol. 2018, No. 1, p. 15903). Briarcliff Manor, NY 10510: Academy of Management.

[25] Margolis, J. (2017). Stuck in the shallow end, updated edition: Education, race, and computing. MIT press.

[26] Connolly, R. (2020). Why computing belongs within the social sciences. Communications of the ACM, 63(8), 54-59.

[27] Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. Education and information technologies, 22, 469-495.

[28] Petri, G., & von Wangenheim, C. G. (2017). How games for computing education are evaluated? A systematic literature review. Computers & education, 107, 68-90.

[29] Zawacki-Richter, O., & Latchem, C. (2018). Exploring four decades of research in Computers & Education. Computers & Education, 122, 136-152.

[30] Oyelere, S. S., Suhonen, J., Wajiga, G. M., & Sutinen, E. (2018). Design, development, and evaluation of a mobile learning application for computing education. Education and Information Technologies, 23, 467-495.

[31] Broisin, J., Venant, R., & Vidal, P. (2017). Lab4CE: a remote laboratory for computer education. International Journal of Artificial Intelligence in Education, 27, 154-180.

[32] Bulman, G., & Fairlie, R. W. (2016). Technology and education: Computers, software, and the internet. In Handbook of the Economics of Education (Vol. 5, pp. 239-280). Elsevier.

[33] Frezza, S., Daniels, M., Pears, A., Cajander, Å., Kann, V., Kapoor, A., ... & Wallace, C. (2018, July). Modelling competencies for computing education beyond 2020: a research based approach to defining competencies in the computing disciplines. In Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (pp. 148-174).

[34] Adeshola, I., & Adepoju, A. P. (2024). The opportunities and challenges of ChatGPT in education. Interactive Learning Environments, 32(10), 6159-6172.

[35] Fitria, T. N. (2021, December). Artificial intelligence (AI) in education: Using AI tools for teaching and learning process. In Prosiding seminar nasional & call for paper STIE AAS (pp. 134-147).

[36] Chan, C. K. Y. (2023). A comprehensive AI policy education framework for university teaching and learning. International journal of educational technology in higher education, 20(1), 38.

[37] Alam, A. (2021, November). Possibilities and apprehensions in the landscape of artificial intelligence in education. In 2021 International Conference on Computational Intelligence and Computing Applications (ICCICA) (pp. 1-8). IEEE.

[38] Grassini, S. (2023). Shaping the future of education: exploring the potential and consequences of AI and ChatGPT in educational settings. Education Sciences, 13(7), 692.

[39] Shi Keke. (2024). Research on Optimization Path of Higher Education Management Based on Collaborative Filtering Algorithm. International Journal of High Speed Electronics and Systems (prepublish).

[40] Eerdenisuyila Eerdenisuyila,Hongming Li & Wei Chen. (2024). The analysis of generative adversarial network in sports education based on deep learning.. Scientific reports(1),30318.

[41] Junwei Xu & Bin Liu. (2025). Enhancing self-mixing interferometry sensing signal by cycle-consistent generative adversarial network. Optics and Laser Technology112431-112431.

[42] Yu Hao. (2022). Retraction Note: Platform Design of Sports Meeting Management System for Regular Colleges and Universities Based on B/S Structure. Wireless Personal Communications(2),1517-1517.