Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 18 No. 7 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijim.v18i07.48305

PAPER

Research on the Construction and Application of an Intelligent Education Learner Model Based on the UTAUT Theory

Gao Qing(✉)

School of Education, Hefei University, Hefei, China

gaoqinggq@outlook.com

ABSTRACT

With the rapid development of computer technology, a new educational model, the innovative education learner model, has emerged as a product of the deep integration of technology and education. In this paper, we will begin by organizing the theories and models related to technology acceptance. We will select the UTAUT model, known for its high explanatory power, as the theoretical framework. Subsequently, we will comprehensively analyze the dataset and conduct in-depth habit mining. The effectiveness of applying K-means concepts to address the classification of clusters of learners' learning habits is confirmed. The feasibility of the LSTM algorithm in predicting learners for exercise responses is also demonstrated. Next, a learning cluster construction method based on intelligent learner clustering is proposed. The methods of MDS+K-means and spectral clustering are selected for clustering. Learning clusters are constructed, and the performance of the two types of algorithms is compared and analyzed. Finally, the enhanced text feature extraction algorithm is utilized to design and implement the corresponding system for the practical application of the innovative educational learner model. The final experiment proves that the text features extracted by the model are effective, with an error rate of only about 2.8%, thus demonstrating that the intelligent educational learning model in this paper is reasonable.

KEYWORDS

learner model, smart education, mobile learning, UTAUT, k-means, text feature extraction algorithm

1 INTRODUCTION

Education informatization has entered a new stage of development. The shift is moving from digital education to practical education, supported by the utilization of big data analysis, artificial intelligence, and other cutting-edge information technologies. With factors such as personalized services, insightful commentary,

Qing, G. (2024). Research on the Construction and Application of an Intelligent Education Learner Model Based on the UTAUT Theory. *International Journal of Interactive Mobile Technologies (iJIM)*, 18(7), pp. 139–157. https://doi.org/10.3991/ijim.v18i07.48305

Article submitted 2024-01-04. Revision uploaded 2024-02-02. Final acceptance 2024-02-02.

natural interaction, and ubiquitous access, intelligent education has become a trend in enhancing educational informatization in China and a fundamental step in implementing the directives of the nineteenth-party congress on "accelerating education modernization." The interactive function of an online learning platform includes two aspects: one is the interaction among learners, and the other is the interaction between learners and the learning content. The interactive tools on the online learning platform mainly include a course platform, a social networking platform, teaching tools, and collaboration tools. Those serve various functions, such as course resources and course management in the course platform, facilitating teacher-student interaction and collaboration in the social networking platform, supporting teaching design and enabling implementation in the teaching tools, and collaborative discussions and exchanges in the collaboration tools. Other support resources on an online learning platform include interactive functions, course materials, and additional learning resources such as videos, audios, images, and documents. These learning materials are closely related to the course resources in terms of form while also offering learners varying degrees of assistance and support.

The modeling of inexperienced individuals aims to formally describe the personalized traits of novices. This enables machines to promptly recognize newcomers and provide tailored services to learners. The IEEE Learning Technology Standards Committee has proposed a learner model that includes aspects such as personal information, learner preferences, learner safety, learner performance, and learning relationships.

The investigation into learner models primarily concentrates on theoretical research, and the components included in the learner models are comprehensive and encompass various aspects [1–3]. However, they are difficult to observe in tangible progress and are not very feasible. In contrast, overseas research has focused on the positive traits of newcomers and has been developed and applied in practice. In terms of knowledge models, Armendariz et al. proposed a learner-understanding model based on the connections between domain concepts. Chaplot et al. suggested a learner model that automatically modeled the relationships between extraordinarily different domain concepts. Guerra et al. utilized the concept of networks to model learner knowledge. Huang et al. focused on the integration of skills, skill-to-skill relationships, and skill-to-item relationships in modeling learners' knowledge. In terms of cognitive ability, Mejia et al. proposed a learner model to estimate the cognitive deficits of learners with dyslexia or reading difficulties. In terms of mastering styles, the mainstream theories for understanding fashion include VARK and Felder-Silverman learning style theory [4–5]. Elbishouty et al. explored the software of Felder-Silverman, studying patterns in the design of an online course, and developed a course evaluation tool based on course feedback data using a linear regression model. The device allows instructors to determine the degree to which an instruction supports a specific learning style based on the Felder-Silverman model.

2 LEARNING HABIT MINING MODEL FOR LEARNERS

2.1 Modelling learner habits

Learners have multiple categories of learning habits with several values under each category. This section models the learning habits of learners using a vector space model. The definition of learning habits in the VSM is shown in the following equation:

$$\begin{cases} V = \left\{ \overrightarrow{P_1}, \overrightarrow{P_2} \ldots \overrightarrow{P_k} \right\} \\ \overrightarrow{P_j} = \left\{ c_{j1}, c_{j2} \ldots c_{jn} \right\} \\ c_{ji} \in C_i, i \in [1, n] \end{cases} \tag{1}$$

Where V is the vector space of learning habits for all learners, $\overrightarrow{P_j}$ denotes the learning habit vector of the jth learner, n is the number of learning habit categories, C_i is the value domain of the i-th learning habit category, and C_{ji} is the i-th specific learning habit of the j-th learner.

2.2 Learning habit-based learner grouping

The grouping of learning habits is based on the learning habits of learners, i.e., learners with the same or similar learning habits are grouped into the same groups.

Description of the learning habit grouping problem. The primary objective of this initiative is to group novices into communities with similar interests based on their learning behaviors to enhance future friend suggestions, support sharing, and other activities. The clustering challenge can be conceptualized as a clustering problem that falls under unsupervised learning and involves the utilization of the enhanced K-means algorithm [6].

K-means defines a set s of learners, a set P of habitual features of learners, and a vector $\vec{P} = (p_1, p_2, p_3 \ldots p_n)$ of learning habitual features of learners. It $p_i \in P$ clusters all habitual features of learners as a set of vectors using the K-means algorithm to minimize its cost function. The cost function is provided in Equation (2).

$$J(c, \mu) = \sum_{i=1}^{m} \| x^{(i)} - \mu_{c^{(i)}} \|^2 \tag{2}$$

Where m – the number of sample points.

$x^{(i)}$ – the vector of learning habit features of the ith sample.

$\mu_{c^{(i)}}$ – the plurality of the cluster to which the ith sample belongs.

The more similar the learning habits of the learners in each cluster, the smaller the value of the error sum of squares.

Learning habit cluster segmentation algorithm. In this section, the attributes of the learners' habits are discrete values, and the distance between the values has no real meaning but can only be described as 'similar' or 'dissimilar.' Therefore, some modifications to the original calculation method are required when determining the distance between two students. The algorithm for calculating the similarity distance is as follows [7–8]:

```
  d←0
  for an attributes in node, do
   if node attribute! = c.attribute do
        d←d+1
   end if
  end for
return sqrt(d)
```

The main idea of comparing similarity distances is to assess the number of shared attributes between two points. The more precise the details, the more similar the two are. The center, denoted as *c*, of the cluster is not necessarily a real learner, as it is determined by the majority of each attribute in the current cluster and may consist of a combination of 'virtual learners.'

Based on the learning habits of the learners, K-means is used to cluster the learners in the following way:

```
Center←Random(k)
    j←0
    while j<J do
        j←0
        for students inhabit, do
            min_d←Max
            for c in Center, do
                d←Similarity_distance(student,c)
                If d<min_d do
                    min_d←d
                    student c←c
            end for
            j←j+min_d
            S[student]←student_c
        end for
        Center←Computing_Center(Center)
    end while
    return S
```

Evaluation criteria. As the algorithm is based on the idea of K-means, it is natural that the evaluation criteria are selected with appropriate references [9–11].

The first is the sum of squared errors, which is calculated as shown in equation 1.

$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} \left| p - m_i \right|^2 \tag{3}$$

Where C_i is the ith cluster, p is the sample points, C_i, m_i is the center of mass of C_i (the mean of all samples in C_i, in this paper, the plurality of all samples), and SSE represents the clustering error of all samples, indicating the effectiveness of the clustering process. As the number of clusters increases, the samples will be divided more finely, and the degree of aggregation of each set will progressively increase. Consequently, the error squared and SSE will naturally become smaller gradually. Moreover, when the number of clusters (k) is much less than the actual range of clusters, the decline in SSE will be significant. This is because increasing k will substantially increase the degree of aggregation of each cluster [12]. The declines in SSE will plummet and then level off as the price of 'ok' continues to increase. In other words, i.e., the graph of SSE versus 'ok' resembles an elbow shape, and the value of 'ok' corresponding to this elbow represents the optimal number of clusters in the data. Therefore, this method of judgment is also known as the elbow method [13].

Another evaluation criterion is the contour coefficient. The contours coefficient of a specific sample point X_i is defined as shown in Equation (4).

$$S = \frac{b - a}{max(a,b)} \tag{4}$$

Where "*a*" represents the average distance between the sample X_i and the other samples in the same cluster, known as cohesion, and "*b*" represents the distance between the sample X_i and all the samples in the nearest cluster, known as separation. The nearest cluster is defined in Equation (5).

$$C_j = \arg\min_{C_k} \frac{1}{n} \sum_{p \in C_k} \left| p - X_i \right|^2 \tag{5}$$

P is the sample in a certain cluster C_k. The average contour coefficient is calculated by determining the contour coefficients of all samples and then averaging them. The average contour coefficient has a range of [–1,1]. The closer apart they are within a cluster, and the further the samples are between clusters, the larger the average contour coefficient and the better the clustering effect. Then, it is natural that the k with the most significant average contour coefficient is the best number of clusters.

2.3 Prediction of exercise scores based on learning habits

The process of observing a learner's behavior over a period of time after logging in is, indeed, a time series. It is feasible to predict whether a learner will provide the correct response to a specific question based on the pattern of learning behaviors exhibited during the task. This will be used as a parameter in calculating weights in the process of planning direction in getting-to-know activities. Typical time sequence evaluation algorithms include recurrent neural networks, long short-term memory networks (LSTM), etc. [14].

(1) Recurrent neural network is a type of neural network primarily designed for processing time-series data samples. Each layer not only outputs to the next layer but also generates a hidden state for the current layer to utilize when processing the next pattern (see Figure 1).

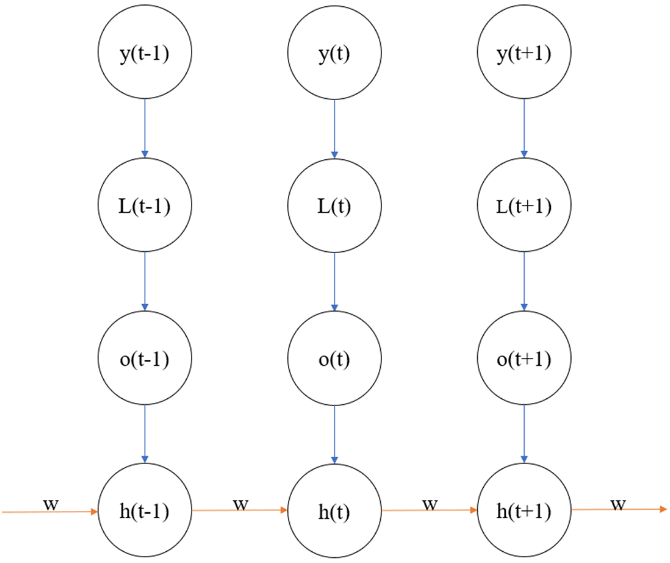

Fig. 1. RNN network structure

(2) Long short-term memory network is a modified version of the RNN, which addresses the issue of gradient dispersion. The original RNN has a single state, h, in its hidden layer, making it highly sensitive to short-term input. In contrast, the LSTM enhances the neuron by introducing a memory cell, c, to store long-term information.

This design allows the current state to be influenced by past input. A diagram of the LSTM network is shown in Figure 2.

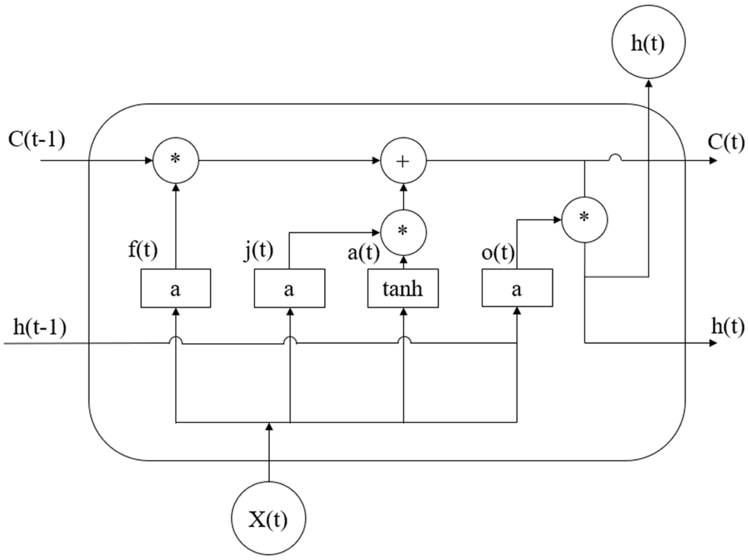

Fig. 2. LSTM network structure

The LSTM model consists of has three control gates within a single neuron: the input gate, the output gate, and the forget gate.

Forgetting gate: Selectively forgetting some information from the past. Here, the past memory information passed to the cell state is defined as in Equation (6).

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{6}$$

Input gate: selectively remembers certain information from the current input; in this case, the current state information passed to the cell is defined as in Equation (7) and Equation (8).

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{7}$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{8}$$

State update: The LSTM was proposed to address the issue of RNN gradient dispersion by transforming the states in the RNN from cumulative to cumulative, ensuring that the state C, C_t passed to the next time step is computed as specified in Equation (9).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{9}$$

Output gate: The output gate calculates the output state at the current moment and transfer the current output to the next moment. The output is determine according to Equation (10) and Equation (11).

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{10}$$

$$h_t = o_t * \tanh(C_t) \tag{11}$$

From the definition of output, it follows that by taking a sequence of a learner's learning behaviors as input, an output vector can be obtained to predict the learner's answer to a test question [15].

3 A METHOD FOR BUILDING SMART EDUCATION LEARNING CLUSTERS BASED ON LEARNER CLUSTERING

In this section, we present an approach for constructing learning clusters in online training platforms, providing an automated solution for building learning clusters in a learner-centered intelligent training model, and facilitating subsequent learning resource recommendation and sharing among learners [16–17]. Firstly, an illustration of learner traits and studying clusters is provided. Then, a clustering evaluation is conducted between learners according to two potential clustering strategies to organize the corresponding learning clusters. Further, this section quickly introduces the dataset used in this study, presents the experimental results of two types of clustering algorithms on real data and the process of parameter tuning, and finally compares and discusses the overall performance of the two algorithms.

3.1 Smart education learner model clustering algorithm design

Cluster construction algorithm based on MDS and k-means. The characteristic vector illustration of every learner incorporates high-dimensional data. To facilitate the clustering of similar learners, the feature vector can be reduced to a two-dimensional space through dimensionality reduction. When the similarity, i.e., the distance, between each of the two objects is established, the representation of these objects in the reduced-dimensional space is determined and adjected to match the original similarity as closely as possible, minimizing any distortion caused by the dimensionality reduction. This dimensionality reduction is computationally efficient and more easily visualized [18].

After obtaining a representation of the learner's features in two dimensions, the K-means algorithm is executed to cluster similar learners by spatial distance. The flowchart for the execution of the K-means algorithm is depicted below (see Figure 3).

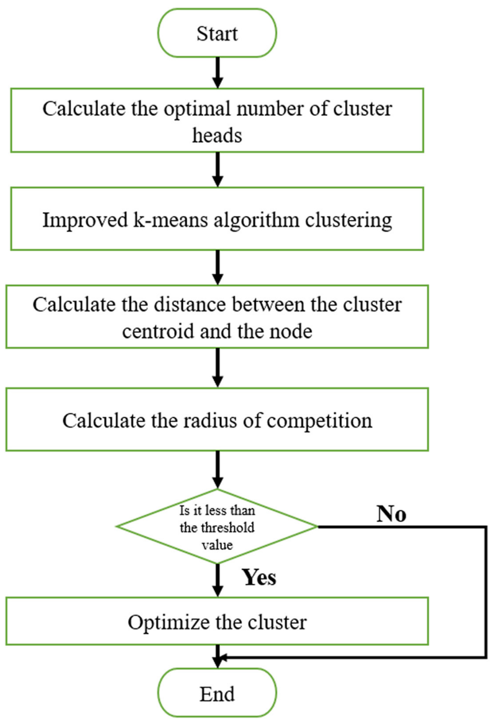

Fig. 3. Flowchart of K-means clustering algorithm

The pseudo-code for the K-Means algorithm is as follows:

```
Centers←initial k centers
  while (True):
        for i=1 to len(Dataset)do:
              C(i)←min (distance (Dataset,Centers))
        end for
  if C(i) –Cold<threshole
         break
   for j=1 to k do:
           Centers(j)←mean (data assigned cluster j)
C(i) = Cold
end while
return C
```

Cluster construction algorithm based on spectral clustering method. The spectral clustering algorithm first constructs the similarity matrix S for the set of input sample points. First, it constructs a similarity matrix. Then, it obtains the adjacency matrix and degree matrix from the similarity matrix. After that, it calculates the Laplacian matrix, normalizes it, and finds the eigenvalues and eigenvectors. Next, it performs normalization again and creates the eigenmatrix. Finally, it applies the specified clustering method to each row using the eigenmatrix.

The result of spectral clustering is obtained by applying the specified clustering method to each row. In this experimental spectral clustering algorithm, the similarity matrix S is constructed based on the distances between the spatial data points, and the fully connected method is used to construct the adjacency matrix W [19]. The challenge is that the e-nearest neighbor technique is not accurate enough to measure distances, which can also impact clustering outcomes. Additionally, the k-nearest neighbor technique needs to address the problem of the adjacency matrix not being a symmetric array. In the fully connected method, a Gaussian kernel function is used to define the edge weights as the distance, calculated as shown in Equation (12), where x_i and x_j are any two points in the set of sample points.

$$K(x_i, x_j) = \exp(-\gamma \,||\, x_i - x_j \,||^2) \tag{12}$$

The degree matrix D is a diagonal matrix, and the Laplacian matrix is calculated as $L = D - W$. Furthermore, L is a symmetric matrix, which implies that thus, its eigenvalues are real numbers. Next, the spectral clustering algorithm divides the graph G into k subgraphs. Here, V represents the set of points in the graph, and E represents the set of edges. The set V_1, V_2, ..., V_k of points, demoted as of the mutually disconnected subgraphs of $G(V,E)$ satisfies equations (13) and (14).

$$V_i \cap V_j = \varnothing, \forall V_i, V_i \in V \tag{13}$$

$$V_1 \cup V_2 \cup \ldots \cup V_k = V \tag{14}$$

The pseudo-code for the polyglot class algorithm is as follows:

for *data_i,data_j* **in** Dataset:
    S←Euclidean_distance(*data_i,data_j*)
end for
W←Gaussian(γ)
L←D–W
L_s←normalized(L)
value←eigenvalue(L_s,k)
V←eigenvector(*value*)
while $C^{(i)} - C^{old} > \varepsilon$:
    $C^{(i)}$ = K-Means(V,k)
  end while
  return C

3.2 Analysis of experimental results and optimization of MDS+K-Means algorithm

The learner feature matrix was reduced to a two-dimensional space using the MDS dimensionality reduction method, and the resulting feature representations are presented in Table 1.

Table 1. Representation of learner characteristics after dimensionality reduction

User id	MDS_0	MDS_1
25	0.159331	1.890221
33	–0.960107	1.710549
116	0.518119	0.059241
…	……	……
11266	–0.863824	–1.128513
11301	0.331233	1.154126
11302	–0.191632	–1.413861
…	……	……

After the preprocessed, raw data is reduced to two dimensions using MDS, the data is distributed as shown in Figure 4.

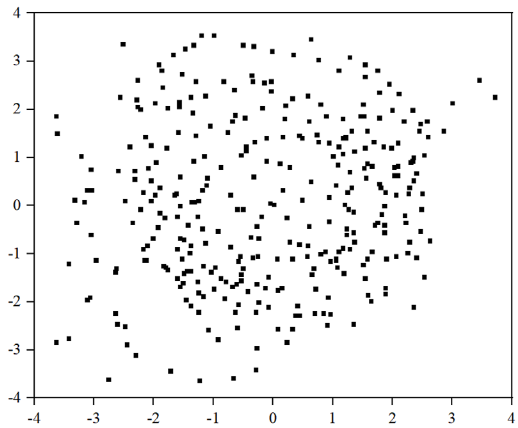

Fig. 4. Distribution of MDS data after dimensionality reduction

The consequences of the experiments are as follows: the information has been clustered using the K-means algorithm, and the classes have been established based on the number of route classes, i.e., k = 23 (see Figure 5).

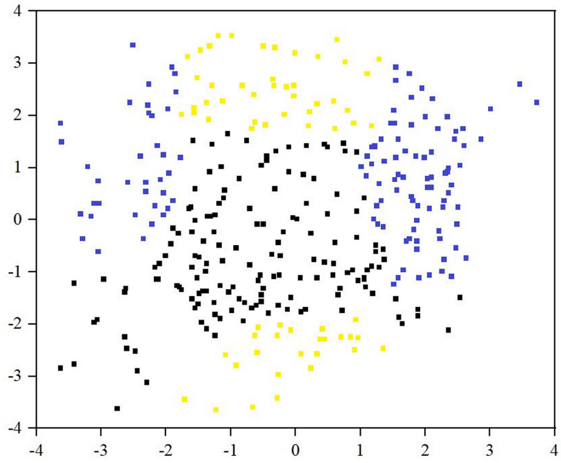

Fig. 5. K-means clustering results (k = 23)

As the number of sample points in each class decreases with increasing k, the aggregation of each class tends to be tighter overall, and the contour coefficients and Calinski-Harabasz scores tend to increase with increasing k [20–21]. We selected the k value corresponding to the clear inflection point as the number of categories in this experiment. We found that k = 30 is the optimal parameter because a smaller k value would result in a less distinct ratio of intra-class to inter-class distances, while a larger k value would lead to an excessive number of categories and could compromise the quality of the clusters. The Harabasz score value is 1464.9.

3.3 Analysis of experimental results and optimization of the spectral clustering algorithm

The results obtained from the spectral clustering are presented in two dimensions. The outcomes depicted in Figures 6 and 7 follow a similar approach to those validated in the K-means clustering algorithm. The clustering results are intuitively organized by ring.

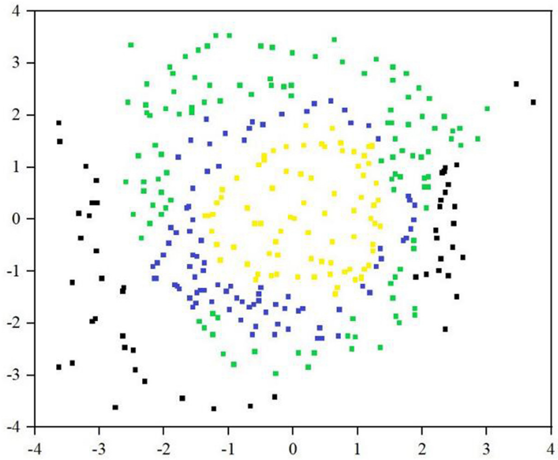

Fig. 6. $\gamma = 0.3$ visualization of the results of the corresponding spectral clustering algorithm

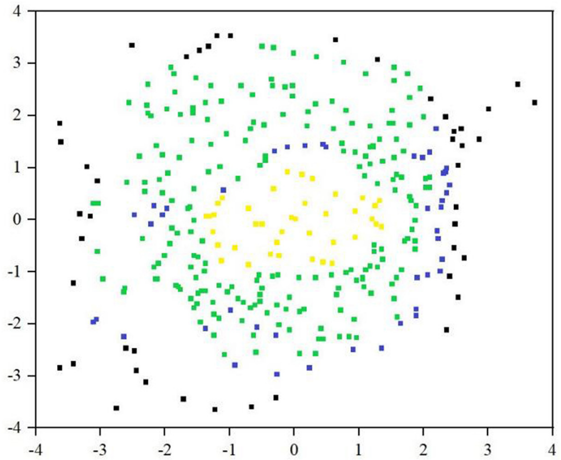

Fig. 7. Visualization of the results of the spectral clustering algorithm for $\gamma = 1$

The profile coefficients tend to increase and then decrease as the value increases, while the Calinski-Harabasz score tends to decrease as the y value increases. Therefore, the optimal value should strike a balance between the scores of the two evaluation criteria as much as possible.

4 UTAUT-BASED ALGORITHM FOR PREDICTING LEARNERS' LEARNING INTERESTS AND INTENTIONS

4.1 Text feature extraction algorithm

TF-IDF (term frequency-inverse document frequency) is a technique for quantifying words in a document. Since the importance of different words in a document or corpus varies, we can calculate the weight of each word using TF-IDF techniques [22–23]. TF-IDF has the following important characteristics: The more often a phrase appears in a specific type of text, the better it is at distinguishing the content of that type of text; if a phrase appears in all types of text, it is poor at distinguishing the content of the text. The TF-IDF algorithm takes into account the frequency of phrases in the document and the distribution of phrases in the average prediction database. However, the wide variety of occurrences of a phrase in the report has too many effects on the standard weight; it no longer considers the distinction in the distribution of phrases between unique instructions.

In this paper, the word2vec algorithm is also utilized to extract phrase vectors. These vectors are then transformed into textual content vectors after appropriate processing and are used for model training along with the features extracted through TF-IDF. Word2ec is a shallow, two-layer neural network model for generating word vectors.

Word2ec is a shallow, two-layer neural network model for generating word vectors. It can efficiently obtain word vectors from a specific corpus using two optimized training models [24]. The CBOW model is based on a one-hot encoded central word and 2n+1-word vectors of the n preceding and following words, and then outputs the word vectors of the central word. The skip-grams model functions in contrast to the continuous bag-of-words model. It involves one-hot encoding of the central word and then generating the word vectors of the n words before and after the central word.

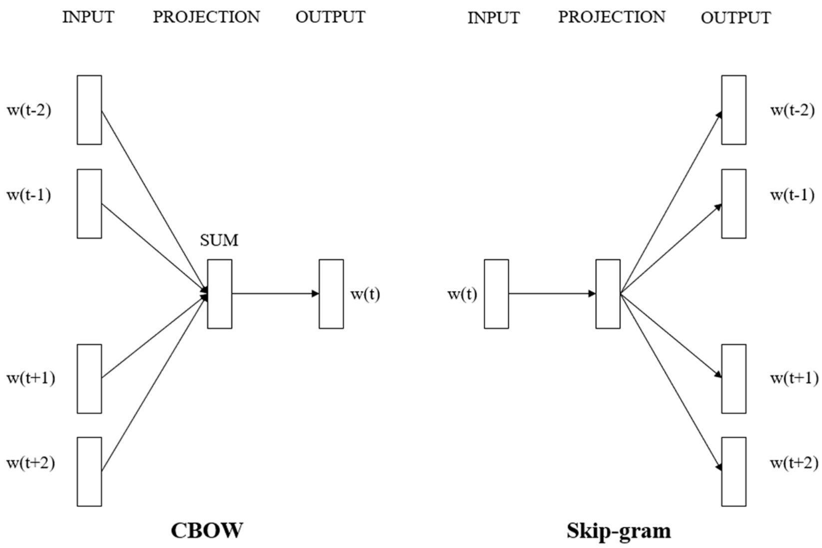

Fig. 8. CBOW model and Skip-gram model for word2vec

As shown in Figure 9, we consider all the words in the corpus as leaf nodes and construct a Huffman tree as the output based on the frequency of the words as weights. The hierarchical SoftMax for the path from the root node to the specified leaf node uses this path to calculate the probability of the specified word rather than using a flat structure. An example of hierarchical SoftMax for the CBOW model is shown in Figure 8, where the input layer consists of word vectors representing contextual words. The word vectors are updated continuously as the training process begins with a random representation of the word vectors [25–26]. The projection layer inputs the vectors to the output layer after a simple summation operation. The producion layer outputs the most likely w; since the number of words in the corpus is fixed at n, the above process can be viewed as a multi-classification problem. Given the features, select one of the n classifications.

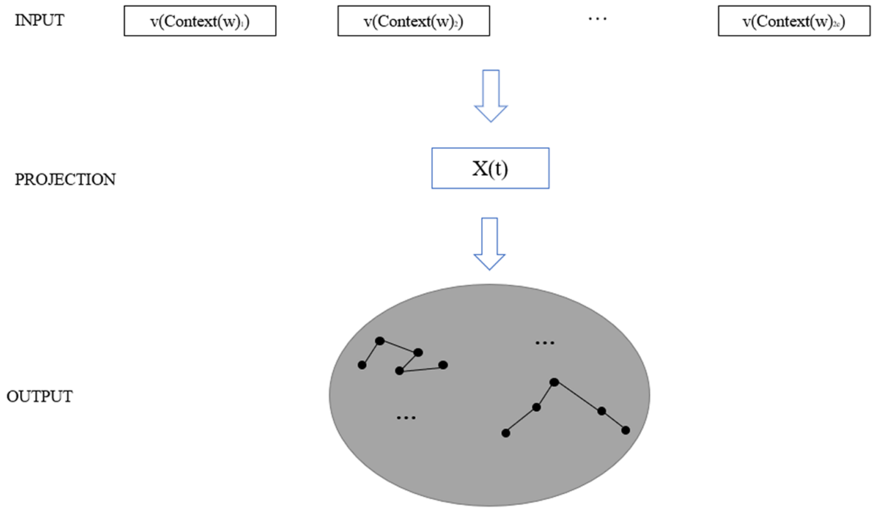

Fig. 9. Schematic of hierarchical softmax optimization of word2vec

Negative sampling (NEG) is a variant of noise-contrastive estimation (NCE). It suggests that a proficient model can differentiate between data and noise using logistic regression. The NEG method involves replacing the central phrase of a sequence of phrases in the corpus with another word. These words, which are no longer present in the corpus, are used as negative samples. This approach aims to optimize the algorithm by maximizing the likelihood of excellent samples while minimizing the likelihood of poor samples. In word2vec, as shown in Figure 10, note that $l_0 = 0, l_k = \sum_{j=1}^{k} len(w_j), k = 1,2,\ldots,N$ the jth word in the corpus is denoted as w_j, with $\{l_j\}_{j=0}^{N}$ as the division node to obtain the interval [0,1] on a non-isometric division, $I_i = (l_{i-1}, l_i), i = 1,2,\ldots, N$ as N division intervals. Further, introduce an equidistant division on the interval [0,1]. Let the division node be $\{m_j\}_{j=0}^{M}$, whose M>>N.

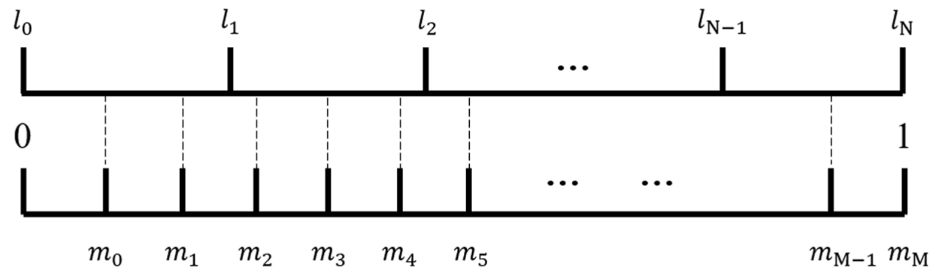

Fig. 10. Schematic of the creation of the word2vec negative sampling Table (·) mapping

By projecting the internal division node $\{m_j\}_{j=1}^{M-1}$ onto the non-isometric division, as indicated by the dashed line in the figure above, a mapping relationship between $\{m_j\}_{j=1}^{M}$ and the interval $\{I_j\}_{j=1}^{N}$ can be established.

$$Table(i) = w_k, m_i \in I_k, i = 1, 2, \ldots, M - 1 \tag{15}$$

With this mapping, the sampling process involves generating a random integer r of $[1, M - 1]$ each time; table (r) then represents a sample and executes Icontext(w) l times in total if w_i happen to be selected in one of the selection processes. If the selection is made again [27].

4.2 Prediction system design and implementation

The functional design of the system is shown in Figure 11.

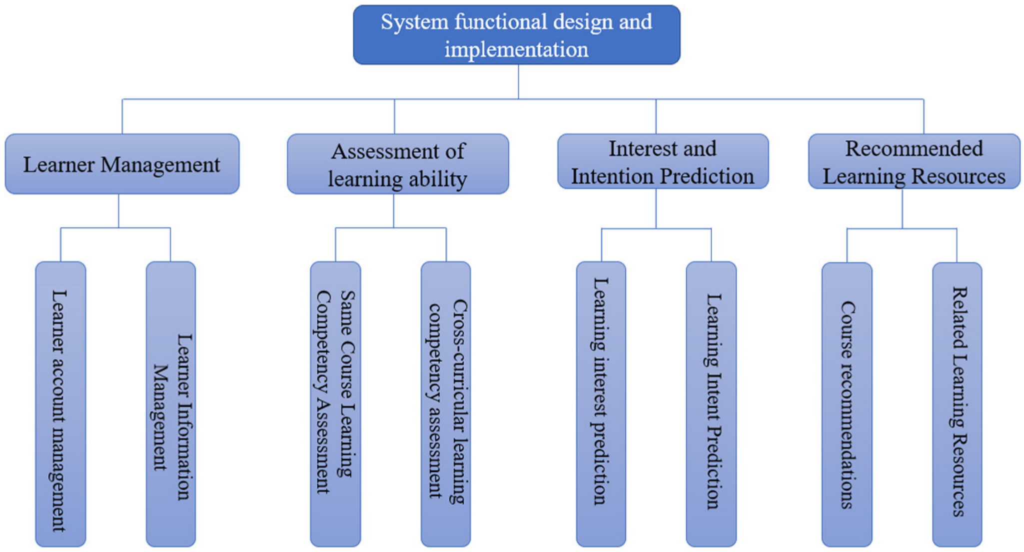

Fig. 11. Functional design of the system for asses

Specifically, the learner learning ability assessment and interest and intention prediction system has the following functions [28]:

1. Learner management. One of the system's primary functions is to enable learners to log in, register, and manage personal information.
2. Learner talent assessment. The algorithm presented in Section 2 is utilized to assess the learner's ability to comprehend, showcasing the learner's studying capability to both the learner and the administrator. It also stores and maintains the learner's capability value, preparing the essential parameters for matching the learning resources when recommending study materials.
3. Learning activity and intention prediction. The algorithm introduced in Chapter Three considers learners' study preferences and goals. It displays beginners their learning activities and current learning objectives while storing and maintaining these activities and objectives as vectors. This process prepares the necessary parameters for matching the subject of learning resources when recommending learning materials.
4. Learning resource recommendation. Combining the results of the learning ability assessment with the learner's interests and intentions, the system recommends courses and learning resources that are both engaging and aligned with the learner's abilities.

The architecture of the system mainly consists of a data layer, a control layer, and an application layer, the structure of which is shown in Figure 12.

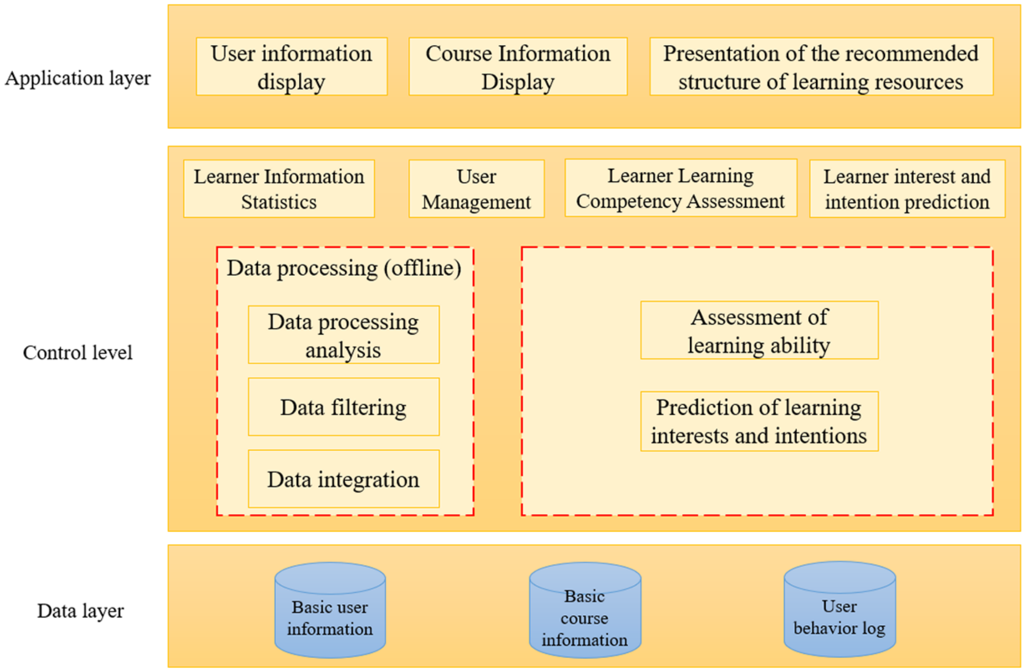

Fig. 12. System architecture diagram

The data layer contains three main types of data: basic user information, basic course information, and user behavior logs.

The control layer contains three main blocks: offline data processing, learning ability assessment, and prediction of learning interest and intention. In terms of data processing, the original data needs to be filtered and integrated, eliminating useless fields so that the data can be correctly input into other modules. Statistical analysis of basic learner information is then conducted based on the results of pre-processing, including statistics on the number of basic learning behaviors and activity analysis [29]. The learning ability assessment section uses the algorithms described in Section 2 to evaluate the learner's capacity for learning. The section on predicting learning interests and intentions utilizes the algorithms introduced in Section 3 to forecast learners' interests and intentions.

The utility layer incorporates a range of features, such as visualizing personal information, displaying the ability to get to know someone, showcasing interests and intentions, presenting route records, and providing suggestion results. The front end of the web page is used to implement and interact with the user, while the operations generated using the utility layer are updated in the system database, and the comments data can be used to optimize the algorithm.

4.3 Analysis of experimental results

The data used is the dataset provided by the school online. After data processing, the dataset used in this section contains 80,000 data points. Each data point comprises learner ID, age, education level, historical search records, withdrawal information, course completion status, and forum comment data. We used the learner's search history, withdrawal information, course completion, and forum comments to train the model to predict the learner's gender (O: unknown; 1: male; 2: female), age (O: unknown; 1: 0–18 years; 2: 19–25 years; 3: 26–30 years; 4: 30–40 years; 5: 41–50 years; 6: 51–99 years), and education (O: unknown education; 1: PhD; 2: Master's; 3: Bachelor's; 4: High school; 5: Middle school; and 6: Primary school. The model was trained using both the TF-IDF algorithm and word2vec word vectors from Section 3.2 to extract text features from user search records [30–31]. Considering that TF-IDF features have high sparsity and dimensionality, which are significantly different from the dense form of word2vec word vectors, only TF-IDF features are combined in the first layer. Then the final model is trained by combining word2vec word vector features in the second layer (see Figure 13).

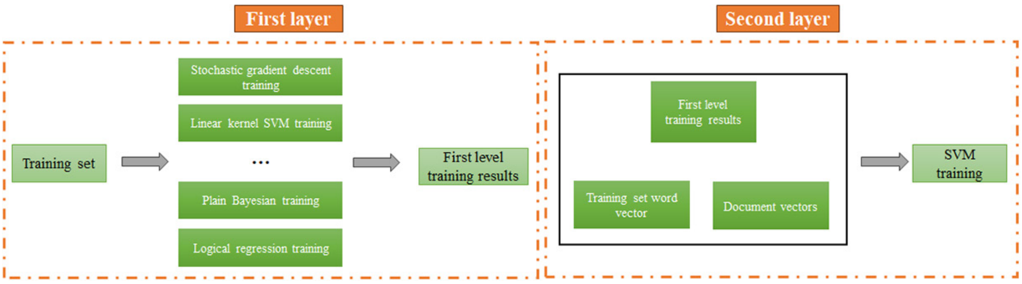

Fig. 13. Model training process

During the model training, there were 18 classification models in the first layer. This contains various types of classifiers and classifiers of the same type with different singular parameters. The reason for incorporating these classifiers was to enhance the training perturbation and address the overfitting issue of the classification model. The SVM classifier is used in the second layer. A file vector matrix, transformed from phrase vectors, is utilized to connect the points with the educational impacts of the first layer. The combined outcomes are then employed for training. The phrase vector points are used to introduce the semantic statistics embedded in the phrase vectors. This is added to the second layer because the sparse nature of the training features in the first layer is not suitable for fusion with word vectors, which have dense characteristics (see Figure 14).

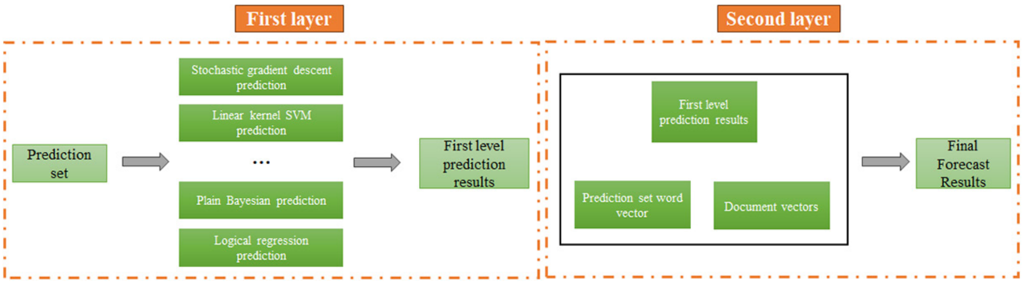

Fig. 14. Model prediction process

The prediction process is similar to the training process, except that the prediction set is used with the trained classifier. The experimental results compared to the benchmark method (text-cnn with only one convolutional layer) are shown in Table 2.

Table 2. Model accuracy comparison

Models	Gender	Age	Academic Qualifications
Text-cnn	0.8327	0.61124	0.6441
Multi-layer fusion model	0.8522	0.6381	0.6793

It can be seen that the feature extraction process accurately extracted the feature words from the learner's behavioral records. Therefore, we can be confident that the final prediction of the learner's interest and intention to learn is relatively accurate.

5 CONCLUSION

Learner personalization systems document the customized traits of students. They can mirror the personalized variations of learners, which is a necessary foundation for adaptive learning systems to provide customized services to learners. This paper proposes a sophisticated training model based on the technological know-how acceptance concept UTAUT, and the learner hobby prediction mannequin illustration and software are studied with statistics and deep mining as the starting point. The particular lookup outcomes are as follows:

1. The learner's learning habits were modeled, the dataset XuetangXDataset was introduced, and six learning habits were obtained by mining the learner's learning habits through processing some of this data. Using the mined learning habits, clustering of learner learning habits and predicting learner scores were conducted. A total of 45 clusters were obtained, and all students were classified into these clusters based on their learning habits. The results will be utilized in future studies. The trained model was used to predict the learners' answers to the questions, and it achieved an average accuracy of nearly 90%.
2. A method for constructing learning clusters based on learner clustering is proposed. Firstly, a formal description of learners and learning clusters is provided. Then, MDS+K-Means and spectral clustering methods are chosen to cluster and

create learning clusters based on the similarity between learners. We clarified the model representation and selected the algorithm. Then, we conducted experimental validation using the pre-processed dataset, XuetangXDataset, which showed an overall better clustering effect. Finally, the parameters were tuned, and the performance of the two types of algorithms was compared and analyzed.

3. This paper proposes a prediction model for studying interest and intention based on text analysis. It collects behavioral data of learners on the learning platform as text data, extracts text features using the extended TF-IDF algorithm and word2vec technology, and designs a corresponding system. Given the challenge of online validation no longer being feasible, an alternative validation approach is suggested to confirm the validity of the textual content points extracted through the model. This, in turn, signifies that the learning activities and goals outlined in this paper are realistic.

6 ACKNOWLEDGMENTS

This study was funded by the University-Industry Collaborative Education Program of the Ministry of Education of China (230902036225457), Teaching and Research Projects of Hefei University (2023hfujyd08), the University Synergy Innovation Program of Anhui Province (GXXT-2023-081), and the Social Science Innovation and Development Research Project of Anhui Province (2023CX177).

7 REFERENCES

[1] G. Liu and F. Wu, "An analysis of the influencing factors of college students' acceptance of mobile learning in Wenzhou: An empirical study based on the extended technology acceptance model," *Modern Educational Technology*, vol. 21, no. 6, pp. 109–114, 2011.

[2] P. Wu, S. Yu, and D. Wang, "Using a learner-topic model for mining learner interests in open learning environments," *Journal of Educational Technology & Society*, vol. 21, no. 2, pp. 192–204, 2018.

[3] D. Armendariz, Z. MacHardy, and D. D. Garcia, "Octal: Online course tool for adaptive learning[C]," in *Proceedings of the First ACM Conference on Learning@ Scale Conference*, 2014, pp. 141–142. https://doi.org/10.1145/2556325.2567849

[4] J. Guerra, Y. Huang, R. Hosseini, *et al.*, "Graph analysis of student model networks[C]," in *CEUR Workshop Proceedings*, University of Pittsburgh, 2015, p. 1446.

[5] Y. Huang, "Deeper knowledge tracing by modeling skill application context for better personalized learning[C]," in *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, 2016, pp. 325–328. https://doi.org/10.1145/2930238.2930373

[6] C. Mejia, S. Gomez, L. Mancera, *et al.*, "Inclusive learner model for adaptive recommendations in virtual education[C]," in *17th International Conference on Advanced Learning Technologies (ICALT)*, IEEE, 2017, pp. 79–80. https://doi.org/10.1109/ICALT.2017.101

[7] W. Chen, C. G. Brinton, D. Cao, *et al.*, "Early detection prediction of learning outcomes in online short-courses via learning behaviors," *IEEE Transactions on Learning Technologies*, vol. 12, no. 1, pp. 44–58, 2018. https://doi.org/10.1109/TLT.2018.2793193

[8] A. Abyaa, M. Khalidi Idrissi, and S. Bennani, "Learner modeling: A systematic review of the literature from the last 5 years," *Educational Technology Research and Development*, vol. 67, pp. 1105–1143, 2019. https://doi.org/10.1007/s11423-018-09644-1

[9] J. Zhang, X. Shi, I. King, *et al.*, "Dynamic key-value memory networks for knowledge tracing," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, 765–774. https://doi.org/10.1145/3038912.3052580

[10] J. Yue and Y. Chen, "Distance learner modeling and personalized learning application based on big data analysis," *China Distance Education*, vol. 7, pp. 34–39, 2017. (in Chinese).
[11] F. Wu, S. Huang, B. Yin, "Study on learner modeling based on scene perception," *Research on Audio-Visual Education*, vol. 40, no. 3, pp. 68–74, 2019. (in Chinese).
[12] R. S. J. Baker, A. T. Corbett, S. M. Gowda, *et al.*, "Contextual slip and student performance prediction after using an intelligent tutor," in *User Modeling, Adaptation, and Personalization: 18th International Conference (UMAP 2010)*, Springer Berlin Heidelberg, 2010, pp. 52–63. https://doi.org/10.1007/978-3-642-13470-8_7
[13] J. I. Lee and E. Brunskill, "The Impact on individualizing student models on necessary practice opportunities," *International Educational Data Mining Society*, 2012.
[14] C. M. Chen, H. M. Lee, and Y. H. Chen, "Personalized e-learning system using item response theory," *Computers & Education*, vol. 44, no. 3, pp. 237–255, 2005. https://doi.org/10.1016/j.compedu.2004.01.006
[15] M. A. Tadlaoui, S. Aammou, M. Khaldi, *et al.*, "Learner modeling in adaptive educational systems: A comparative study," *International Journal of Modern Education and Computer Science*, vol. 8, no. 3, pp. 1, 2016. https://doi.org/10.5815/ijmecs.2016.03.01
[16] J. Nakic, A. Granic, and V. Glavinic, "Anatomy of student models in adaptive learning systems: A systematic literature review of individual differences from 2001 to 2013," *Journal of Educational Computing Research*, vol. 51, no. 4, pp. 459–489, 2015. https://doi.org/10.2190/EC.51.4.e
[17] Z. Jeremic, J. Jovanovic, and D. Ga Evic, "Student modeling and assessment in intelligent tutoring of software patterns," *Expert Systems with Applications*, vol. 39, no. 1, pp. 210–222, 2012. https://doi.org/10.1016/j.eswa.2011.07.010
[18] K. Tourtoglou and M. Virvou, "An intelligent recommender system for trainers and trainees in a collaborative learning environment for UML," *Intelligent Decision Technologies*, vol. 6, no. 2, pp. 79–95, 2012. https://doi.org/10.3233/IDT-2012-0125
[19] C. Y. Ting and S. Phon-Amnuaisuk, "Properties of Bayesian student model for INQPRO," *Applied Intelligence*, vol. 36, pp. 391–406, 2012. https://doi.org/10.1007/s10489-010-0267-7
[20] R. Conejo, M. Trella, I. Cruces, *et al.*, "INGRID: A web service tool for hierarchical open learner model visualization," in *Advances in User Modeling: UMAP 2011 Workshops*, Springer Berlin Heidelberg, 2012, pp. 406–409. https://doi.org/10.1007/978-3-642-28509-7_38
[21] F. Bakalov, I. H. Hsiao, P. Brusilovsky, *et al.*, "Progressor: Personalized visual access to programming problems," in *Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 2011, pp. 205–208. https://doi.org/10.1109/VLHCC.2011.6070400
[22] M. Russell, J. Lippincott, and J. Getman, "Connected teaching and personalized learning: Implications of the National Education Technology Plan (NETP) for adult education," *Draft. Washington, DC: Us Department of Education, Office of Career, Technical, and Adult Education,* vol. 12, p. 2015, 2013.
[23] I. H. Hsiao, J. Guerra, D. Parra, *et al.*, "Comparative social visualization for personalized e-learning," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2012, pp. 303–307. https://doi.org/10.1145/2254556.2254614
[24] P. Brusilovsky, R. Farzan, and J. Ahn, "Comprehensive personalized information access in an educational digital library," in *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, 2005, pp. 9–18. https://doi.org/10.1145/1065385.1065388
[25] S. Bocconi, S. Dini, L. Ferlino, *et al.*, "ICT educational tools and visually impaired students: different answers to different accessibility needs," *Universal Access in Human-Computer Interaction. Applications and Services: 4th International Conference on Universal Access in Human-Computer Interaction (UAHCI 2007)*, Springer Berlin Heidelberg, 2007, pp. 491–500. https://doi.org/10.1007/978-3-540-73283-9_55

[26] C. Marinagi and C. Skourlas, "Blended learning in personalized assistive learning environments," *International Journal of Mobile and Blended Learning (IJMBL)*, vol. 5, no. 2, pp. 39–59, 2013. https://doi.org/10.4018/jmbl.2013040103

[27] P. Brusilovsky, S. Somyurek, J. Guerra, *et al.*, "The value of social: Comparing open student modeling and open social student modeling," in *User Modeling, Adaptation and Personalization: 23rd International Conference (UMAP 2015)*, Springer International Publishing, 2015, pp. 44–55. https://doi.org/10.1007/978-3-319-20267-9_4

[28] Z. Chedrawy and S. S. R. Abidi, "An adaptive, personalized recommendation strategy featuring context-sensitive content adaptation," in *Adaptive Hypermedia and Adaptive Web-Based Systems: 4th International Conference (AH 2006)*, Springer Berlin Heidelberg, 2006, pp. 61–70. https://doi.org/10.1007/11768012_8

[29] F. Colace, M. De Santo, and L. Greco, "E-learning and personalized learning path: A proposal based on the adaptive educational hypermedia system," *International Journal of Emerging Technologies in Learning (Online)*, vol. 9, no. 2, p. 9, 2014. https://doi.org/10.3991/ijet.v9i2.3211

[30] E. Knutov, "Generic Adaptation Framework for unifying adaptive web-based systems," *PhD, Technische Universiteit Eindhoven*, Eindhoven NL, 2012.

[31] S. Sosnovsky and P. Brusilovsky, "Evaluation of topic-based adaptation and student modeling in QuizGuide," *User Modeling and User-Adapted Interaction*, vol. 25, pp. 371–424, 2015. https://doi.org/10.1007/s11257-015-9164-4

8 AUTHOR

Gao Qing graduated from Henan Normal University in 2016 with a master's degree in education. She is currently working in the Research Department of Hefei University, and her research interests are Principles of Education, Curriculum and Instruction, Educational Technology, etc. In the past five years, she has presided over or participated in more than 10 scientific research projects at all levels, published many academic papers, and has won the Anhui Provincial Teaching Achievement Award (E-mail: gaoqinggq@outlook.com).