Hindawi
Mobile Information Systems
Volume 2023, Article ID 9686763, 10 pages
https://doi.org/10.1155/2023/9686763

Research Article

Dynamic Student Data Management Using Resource Optimization Technology in Higher Education Platforms

Chenghao Lu[1,2] and Omer Saeed[3]

[1]*School of Public Policy&Management, China University of Mining and Technology, Xuzhou, China*
[2]*Office of Academic Affairs, Jinling Institute of Technology, Nanjing, China*
[3]*Institute of Management, Kyrgyz International Universal College, Bishkek, Kyrgyzstan*

Correspondence should be addressed to Omer Saeed; omersaeed@mail.cu.edu.kg

Received 3 September 2022; Revised 22 September 2022; Accepted 3 October 2022; Published 12 April 2023

Academic Editor: Chi Lin

This study combines intelligent resource optimization technology to build a dynamic student data management model and suggests a fuzzy hierarchical network representation model based on isomorphism and homogeneity in order to increase the effectiveness of dynamic student data management in colleges and universities. The important semantics of nodes in the network are also captured in this study using fuzzy k-kernel decomposition as a technique for multigranularity partitioning. Based on the SIR model, FHNE compares the production of the sequence to the process of information transmission in the random walk stage, which increases the node sequence's accuracy. According to the research, the dynamic student data management system that is used in the higher education platform that is suggested in this study can significantly increase the effectiveness of managing student data.

1. Introduction

The student model ITS (intelligent teaching system) changes dynamically according to the interaction between students and the system and the learning situation, which provides the basis for the determination of teaching strategies and teaching resources required by the system for further teaching. The student model is currently the most difficult part of ITS design and a hot issue in ITS research [1]. Guided by relevant teaching theories and learning theories, starting from the individual learning situation and learning needs of learners, based on research and analysis of factors affecting learning, a student model of an intelligent teaching system is designed [2]. There are many factors in the design of the student model. For instance, it must be able to accurately describe students' learning situations, work with the system's representation of course information, help the system design new teaching tactics, and help with implementation [3].

The coverage model expresses the domain knowledge that students want to learn and their constraints as a directed knowledge structure graph, the student's learning state is regarded as a subgraph of this graph, and the learning process is regarded as the approximation process of the subgraph to the original graph. The system based on this model can obtain the knowledge structure defects of students according to the comparison between the domain knowledge structure diagram and the students' knowledge state diagram, to recommend the content to be learned to the students [4].

Deviation model: This model records the deviation of the student's problem-solving path from the expert's path. These deviations describe a certain deficiency of the student at this knowledge point and can give specific remedial measures according to the type of deviation [5].

The cognitive model reflects the differences in cognitive ability and cognitive structure of each learner. We can comprehend pupils' initial learning capacity and knowledge structure by examining their cognitive peculiarities. The purpose of analyzing students' cognitive ability and cognitive structure is to formulate effective teaching strategies for specific teaching tasks [6]. Mental model: It refers to a collection of interrelated speech or representation propositions, which is the deep knowledge base for people to make inferences and predictions. Mental models describe and record the psychological changes of students as they learn, as well as their effects on the learning process [7].

Different student models represent different theoretical foundations of learning. In the educational field, behaviorist learning theory has always occupied an important position. In the intelligent teaching system established on the basis of the teaching practice of this teaching theory, the corresponding student model is the covering student model. In this model, to understand the student's learning foundation or the student's learning mastery level, students can be tested, and the test results can be analyzed. Generally speaking, a course consists of multiple units, each unit has multiple subsections, and each subsection contains multiple knowledge points [8]. To test the students' mastery of a course, it can be obtained by weighting the mastery of each knowledge point. The feature of the coverage model is that it only considers the students' right and wrong answers. The correct rate of answering questions is lower than a certain percentage when learning a certain knowledge point. Because there is no analysis of the mistakes made by the students, it is impossible to give targeted learning guidance, and the consequences of students' simple repetition of learning are often repeated mistakes. The consequence of this continuous cycle is often that students lose interest in learning and lose confidence in themselves [9]. It can be seen that the coverage model is only suitable for describing the learning situation of declarative knowledge and simple procedural knowledge, and it is not convenient to describe the learning situation of complex procedural knowledge. An introduction to the bias model addresses the aforementioned issues. In order to identify various error kinds and associated faults, it analyzes, summarizes, and contrasts the errors made by students with the appropriate problem-solving techniques. In this way, students only need to relearn the corresponding weak links, and the learning will be more targeted, and the learning effect will naturally be better [10]. In order to establish a deviation model, it is necessary to diagnose the students' learning process, and domain experts set a number of possible errors for each knowledge point. Every time a student gets a question wrong, it must correspond to one or more of the wrong sets. The cognitive model is an indispensable part of the student model [11].

In order to determine a student's cognitive ability, the question of how cognitive ability is represented must first be addressed. In practical applications, a relatively simple and practical cognitive model for students is established. It is believed that educational goals should include the content of three domains, namely, cognitive ability domain, motor skill domain, and affective domain [12]. Cognitive ability determines the choice of learning content. Those with higher cognitive ability can arrange learning content with greater knowledge difficulty. Psychological factors have an important impact on learning. Personality has an effect on learning effect, according to research on the connection between learner personality and learning effect: (1) Students with tenacious-stable personalities lack empathy and place a greater emphasis on facts than on feelings during learning. The subjects frequently receive high marks. Conversely, learners with sensitive personalities tend to do well in arts subjects. (2) The introverted and extroverted personality characteristics of learners are also related to the learning effect: learners with extraverted personality characteristics tend to ignore academic research and prefer stimulating social activities, so they need stronger stimulation when learning. It is suitable for processing information provided quickly; while, learners with introverted characteristics are more suitable for learning with slower information processing and reflective learning [13]. Studies have shown that learners with a field-dependent learning style are suitable for structured learning methods based on facts, well-structured teaching materials, and clear teaching goals. If they are not, they should carry out an individualized learning method based on learner control and encourage them to maximize their learning such as inquiry, discovery, and discussion [14].

"Mastery degree" indicates the degree of mastery of the knowledge point. The system can set a threshold. If the test score is higher than the threshold, the system will no longer arrange learning; if the test score is higher than the threshold, the system will no longer arrange learning. It will be arranged to continue learning; if it is near the threshold, students will be prompted whether they need to continue to strengthen, and the students will choose [15]. The "error number" indicates the type of error that the student made in the knowledge point test. The system can obtain the corresponding error description information and learning prompt information from the error type table and provide it to the student for targeted learning. The learning strategy library records the learning strategies of each knowledge point in the next round of learning obtained after the decision. The following cycle of learning could involve reviewing some of the material from the most recent unit, or it could involve learning something entirely new. The techniques include selecting the subject matter to be taught, how to deliver information, how to teach, and how to create exam questions [16]. The error type table is a collection of all possible errors in the student's study of the subject. This collection is drawn up by experts in advance. New types can be added according to the actual learning process (this will increase the complexity of the system), or they can remain unchanged. Based on this table, the system calculates the student's learning deviation and gives relevant prompt information. The learning assessment table records the learning results of each knowledge point in the unit after each test, which is the basis for adjusting students' knowledge table and error type table, calculating cognitive ability, and then forming teaching strategies. The psychological status table describes the psychological factors of students during learning, and its initial value can be entered by students when they register to indicate their preferences. During the learning process, the system tracks students' learning behaviors and automatically draws judgments [17].

This study combines intelligent resource optimization technology to construct a dynamic student data management model to improve the management effectiveness of the student data management data model.

2. Fuzzy Hierarchical Network Representation Based on Isomorphism and Homogeneity

Fuzzy k-kernel decomposition is proposed to learn more robust node representations and encode more semantics. To overcome the limitations of the original k-kernel decomposition, FHNE proposes fuzzy decomposition to replace the original strict decomposition and establishes a fuzzy k-kernel decomposition model. In the fuzzy k-kernel decomposition method, the membership function of nodes assigns membership values according to the similarity between node degrees and k values. Therefore, nodes that are very close to the k-kernel will be activated during the fuzzy k-kernel decomposition. The core of this membership function is that the membership value should be positively correlated with the semantics of node importance.

Its model framework is shown in Figure 1.

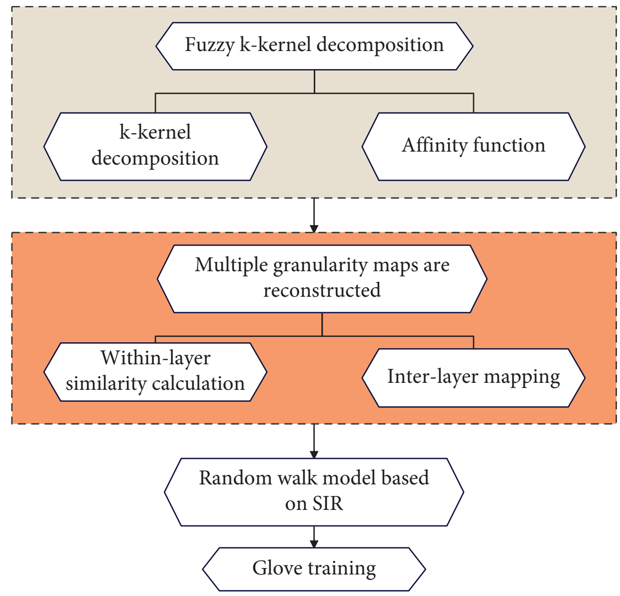


FIGURE 1: FHNE frame diagram.

Based on the k-core decomposition theory, the network can be divided into subnetworks with different degrees of aggregation according to the importance of nodes. The larger the value of k, the stronger the aggregation of nodes. Although k-core decomposition can distinguish the importance of nodes well, it still suffers from the drawback that its "recursive pruning" process is too strict. As shown in Figure 2, if we want to obtain the 2-core subnetwork $H = (C, EIC)$, the condition $\forall v \in C$: $de\,gree_{H(v)} \geq 2$ must be satisfied. Figure 3(a) shows the first pruning from 1-core to 2-core, and Figure 3(b) shows the final 2-core subnet. In the small network shown in Figure 2, the degree of node v is 4, which is relatively important. However, node v will be removed in the process of generating the 2-core network because it does not meet the strict definition of a 2-core network. To better capture the importance of nodes, the strict conditions of the k-kernel decomposition need to be changed by adopting a fuzzy map.

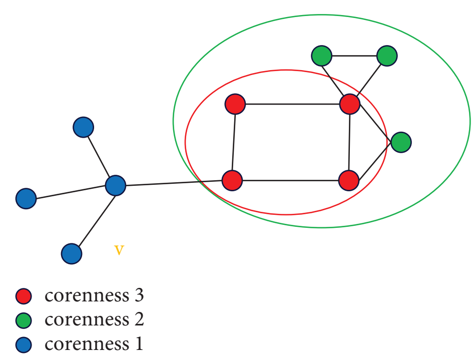


FIGURE 2: Example of k-core.

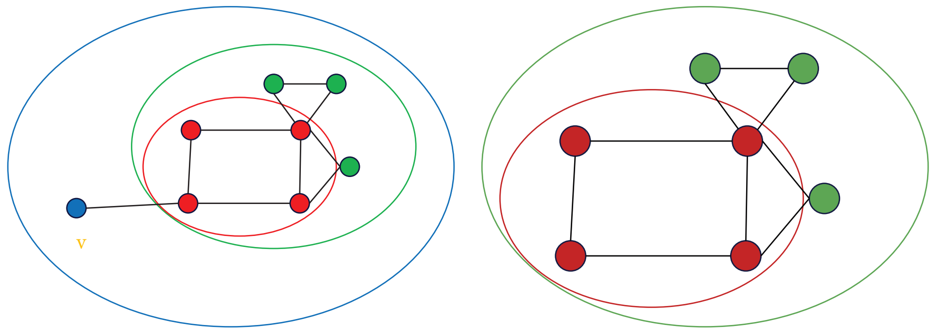
Fuzzy sets are used as a reliable means to quantitatively express inherent ambiguity and uncertainty information and break the boundaries of sets described by characteristic functions. Any element can belong to multiple fuzzy subsets at the same time, and its degree can be represented by a membership value in the interval [0, 1], which is closer to human perception. To solve the problem caused by exact k-kernel decomposition, fuzzy k-core decomposition is proposed.

The key problem of fuzzy systems is the use of membership functions. Generally speaking, fuzzy membership functions are divided into linear and nonlinear. In the FHNE model, the feature of fuzzy k-core decomposition is only node degree. The model hopes that as the value of k increases, the membership degree calculated by the node degree can cover more semantics. It can be seen that the typical characteristics of nonlinear membership functions do not match this, which may lead to a sharp rise or fall in membership. Therefore, the trapezoidal membership function as shown in Figure 4 is adopted, and its corresponding expression is shown in the following formula:

$$\forall v \in G, A(v)_k = \begin{cases} 0, & d(v) < k, d(v) < k, \\ \dfrac{d(v) - k}{b - k}, & k \leq d(v) \leq b, \\ 1, & b < d(v). \end{cases} \quad (1)$$

Among them, $d(v)$ represents the degree of node v, and $A(v)_k$ represents the membership degree of node v belonging to k-core. As shown in Figure 4, node v belongs to k-core if $A(v)_k > \lambda$, where λ is the cutoff set of membership functions.

Definition 1. (Fuzzy k-core decomposition): In the k-core decomposition process, if the membership degree of a node is greater than or equal to the cut set a, the node belongs to k-core, otherwise, it does not belong. If and only if $\forall v \in C$: $A(v)_k > \lambda$ and FH is the largest subgraph with this property, $FH_k = (C, E|C)$ denotes a fuzzy k-core network.



(a) (b)

Figure 3: k-core decomposition process. (a) The first pruning. (b) 2 kernels network.

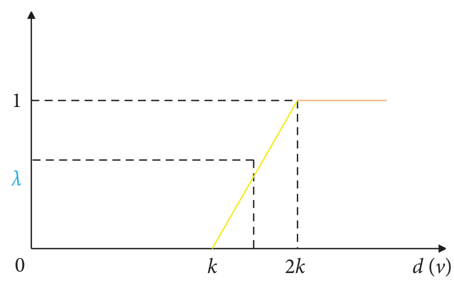


Figure 4: Trapezoidal membership function.

In this section, the original network is divided into multigranularity networks by fuzzy k-core decomposition, and the maximum k value of the network is denoted as max_k. If the step size of fuzzy k-core decomposition is set to 1, it will bring two problems. First, the constructed multigranularity subgraphs will be very large and computationally expensive. Second, the importance of nodes on different granulation layers is not very different, which will lead to an unsatisfactory granulation effect. Therefore, this study defines the step size to control the number of layers l through the fuzzy k-core decomposition algorithm, and the relationship is defined as follows:

$$\eta = \left\lfloor \frac{\max\_k}{l} \right\rfloor. \tag{2}$$

Among them, l represents the number of layers.

Granular computing is a way of thinking from different levels, different perspectives, and different granularities. The original network is divided into multiple subnetworks based on fuzzy k-core decomposition, and each subnetwork with different k-cores represents a granular layer. The larger the value of k, the smaller the subnet size and the stronger the aggregation of nodes in the subnet. To better capture the homogeneity and isomorphism of the network at the same time, this section reconstructs the subnetwork of each granular layer by introducing the structural similarity of nodes. First, the structural similarity between nodes u and j is defined:

$$Sim_k(i,j) = \frac{\min(degree(i), deg(j))}{\max(degree(i), deg(j))}, i, j \in FH_k. \tag{3}$$

Among them, k represents the kth granular layer, and Sim_k represents the structural similarity matrix of nodes. As shown in formula (3), if the degrees of nodes i and j are equal, then $Sim_k(i,j) = 1$. It is worth noting that in the process of building a multigranularity network, the neighborhood information of nodes also needs to be considered. Among them, the adjacency matrix is used to capture the relationship between nodes in each layer to fuse the link information of the network. Therefore, the adjacency matrix of the kth layer can be defined as follows:

$$Adj_k(i,j) = \begin{cases} 1, & (i,j) \in E, \\ 0, & (i,j) \notin E. \end{cases} \tag{4}$$

Combining the above two aspects, two nodes in the same layer are adjacent and have similar importance; then, they will be more likely to get closer representation. The final weight matrix is determined by the adjacency matrix Ad and the structural similarity matrix Sim, and the weight of the kth layer is defined as follows:

$$W_k(i,j) = \delta * Sim_k(i,j) + (1-\delta) * Adj_k(i,j). \tag{5}$$

Among them, δ is the hyperparameter governing the matrices Sim and Adj, and the matrix W can capture the homogeneity and isomorphism of the original network. After reconstructing the multigranularity graph, there is currently no connection between the granular layers, and the core of the mutual mapping between the granular layers is to find similar nodes. Obviously, similar nodes exist not only in the same layer but also in other layers. The larger the value of k, the closer the nodes in the granular layer and the stronger the similarity of nodes. In this section, the weight matrix W is used to calculate the transition probability between granular layers, and it tends to make nodes jump to the granular layer with a larger k value. The interlayer mapping function is expressed as follows:

$$Map_i(k,k+1) = \sum_j \frac{W_k(i,j)}{\left(|V_k| - 1\right)}, \tag{6}$$

$$Map_i(k,k+1) = 1 - Map_i(k,k+1). \tag{7}$$

Map $(k, k+1)$ represents the mapping function from the kth layer to the $(k+1)$ th layer, and Map $(k, k-1)$ is the mapping function from the kth layer to the $(k-1)$ layer, where $|V_k|$ represents the number of nodes in the kth layer. The model establishes a mapping relationship for most of the adjacent granular layers, but the first layer only has the mapping to the second layer, and the last layer only has the mapping to the previous layer. All in all, the interlayer mapping function addresses the connectivity problem of multigranularity graphs.

The first infected user of the SIR infectious disease model corresponds to the seed node of the random walk, and the infection process corresponds to the random walk process. When a node transitions to the removed state, it means that the random walk terminates. Different from the traditional SIR model, the infection probability in this model is no longer a fixed 0, but the similarity between nodes. Similarly, the immune probability is no longer a fixed β, and the immune conditions of nodes are shown in the following formula:

$$W(i,j) < \sum_{v\in V_k} \frac{W_k(i,v)}{\left(|V_k|-1\right)}. \tag{8}$$

In general, the selection of subsequent nodes is only related to the current node. However, the actual situation is that other tail nodes that are already in the walk sequence will also affect the subsequent walk. The model introduces a multinode influence mechanism to allow nodes to consider more information in the process of walking, which is defined as follows:

$$SIMI(i,j) < \prod_{n\in ns[-ls:]} W(i,n). \tag{9}$$

Among them, ns represents the existing node sequence, and ls represents the last few nodes in the sequence ns. Taking into account the time complexity of random walk and the obtained sequence effect, the value of ls is set to half of the glove training window.

This study models the training process of the node sequence. First, some symbols are explained. The matrix of node co-occurrence counts is denoted by X, and X_i represents the number of occurrences of node j in the context sequence of node i, $X_i = \sum_k X_{ik}$ represents the number of occurrences of any node in the context of node i, and $P_{ij} = (Pj/i) = X_{ij}/X_i$ represents the probability of node j appearing in the context of node i.

The same is true for the training of nodes in the network, and nodes can be compared to words. For nodes i and j, this section examines the relationship between nodes by introducing probe node k. The general model can be represented as follows:

$$F(\Phi(i),\Phi(j),\Phi(k)) = \frac{P_{ik}}{P_{jk}}. \tag{10}$$

Among them, $\Phi \in R^d$ is the vector of the node itself, and $\overline{\Phi}\in R^d$ is the vector that is the context of other nodes. The possibilities of the function F are various, but by performing some operations, it can be induced to convert it into an understandable formula. First, F encodes information at a ratio P_{ik}/P_{jk} in the node vector space. Since vector spaces are inherently linear in structure, the most natural approach is to use vector difference. With this goal in the hand, the model can construct F as a function that embodies the node differences, so that formula (10) can be modified as follows:

$$F(\Phi(i)-\Phi(j),\overline{\Phi}(k)) = \frac{P_{ik}}{P_{jk}}. \tag{11}$$

In formula (11), the parameter of F on the left is a vector and the parameter on the right is a scalar. While F can be thought of as a complex function parameterization, such as a neural network, doing so confuses the linear structure one is trying to model. To avoid this problem, the left side of the formula can be dot-product first:

$$F\left((\Phi(i)-\Phi(j))^T\overline{\Phi}(k)\right) = \frac{P_{ik}}{P_{jk}}. \tag{12}$$

This is to prevent F from changing the vector dimension in an incorrect way. It is worth noting that for the node co-occurrence matrix, the roles between the node itself and the context node are interchangeable. To do this, the function needs to satisfy the symmetry, and formula (12) does not satisfy the symmetry, but it can be satisfied by the multistep transformation, and the specific operation is as follows:

$$F\left((\Phi(i)-\Phi(j))^T\overline{\Phi}(k)\right) = \frac{P_{ik}}{P_{jk}} = \frac{F\left(\Phi(i)^T\widetilde{\Phi}(k)\right)}{F\left(\Phi(i)^T\widetilde{\Phi}(k)\right)}, \tag{13}$$

$$F\left(\Phi(i)^T\widetilde{\Phi}(k)\right) = P_{ik} = \frac{X_{ik}}{X_i}. \tag{14}$$

The left side of the formula represents subtraction and the right side represents division. We can execute exponential operations on the left and right sides to transform them into equivalent expressions, that is, $F = \exp$.

$$\Phi(i)^T\widetilde{\Phi}(k) = \log(P_{ik}) = \log(X_{ik}) - \log(X_i). \tag{15}$$

Without the term b on the right-hand side, (15) satisfies exchange symmetry. Although this term is independent of k, it can be absorbed into the bias term b_i of $\Phi(i)$. Then, $\widetilde{\Phi}(k)$ adds an extra bias term $\overline{b}_k$, and the formula restores symmetry.

$$\Phi(i)^T\widetilde{\Phi}(k) + b_i + \overline{b}_k = \log(X_{ik}). \tag{16}$$

Formula (16) is a simple visualization of formula (10). Since the logarithmic function is meaningless when its argument is zero, one way to solve this problem is to add a term to the logarithm, such as $\log(X_{ik}) \longrightarrow \log(X_{ik}+1)$, which guarantees the sparsity of X and avoids the divergence problem of the logarithm. Next, formula (16) is converted into a least squares problem, and the model can learn the node sequence. At the same time, the weight function $f(X_{ij})$ is introduced into the loss function, and the model is defined as follows:

$$\Phi(i)^T\widetilde{\Phi}(k) + b_i + \overline{b}_k = \log(X_{ik}). \tag{17}$$

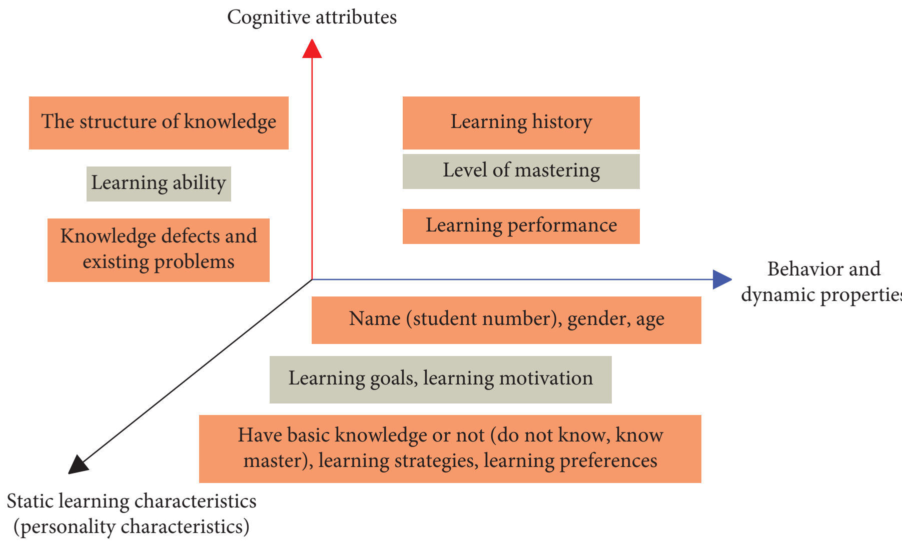


FIGURE 5: 3D model of learner feature attributes.

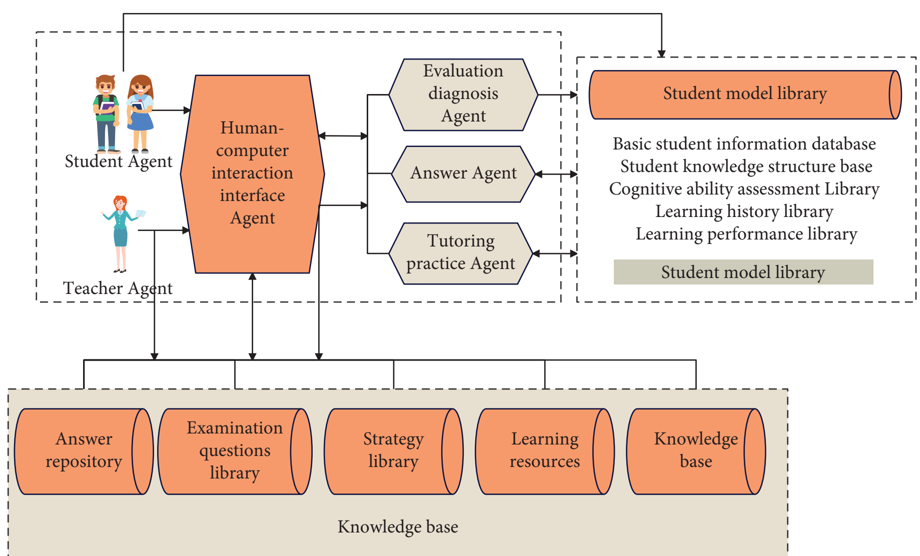


FIGURE 6: Student model library.

Among them, IV is the set of network nodes. If the same weight is used for each co-occurring node, the important difference between nodes cannot be reflected. Therefore, this section introduces the weight function $f(X_{ij})$, which needs to meet the following properties:

(1) $f(0)=0$. When the number of co-occurrences of words is 0, the corresponding weight should be 0.

(2) F must be a nondecreasing function, so as to ensure that the weight will not decrease as the number of node co-occurrences increases

(3) For large values of x, the upward trend of f(x) should be relatively small, so that the weight of frequently occurring co-occurring nodes is not too large

Combining the above three points, the definition of the weight function f(x) is as follows:

$$f(x) = \begin{cases} \left(\dfrac{x}{x_{\max}}\right)^{\alpha}, & if\, x < x_{\max}, \\ 1, & otherwise. \end{cases} \tag{18}$$

The performance of the model depends to a small extent on the cutoff value. This model fixes it as $x_{\max} = 100$ for all experiments. According to expert experience and many experiments, the model effect is improved to a certain extent when $\alpha = 3/4$.

3. Dynamic Student Data Management

The model is constructed according to the basic information of the learner, and it is simplified into a three-dimensional student model as shown in Figure 5.

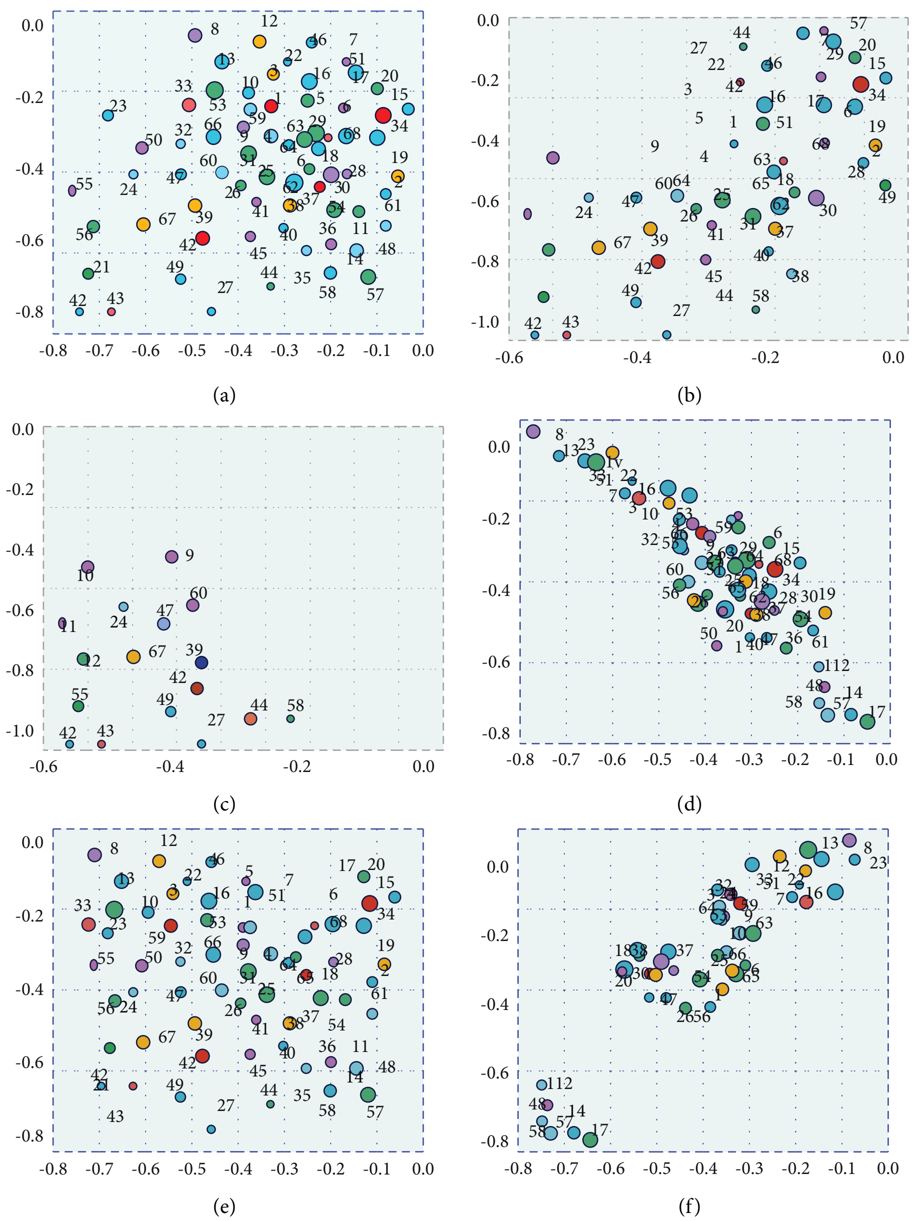


Figure 7: The spatial visualization of the network vector of the students' teaching process. (a) Deep walk. (b) Node2vec. (c) SDNE. (d) Struc2vec. (e) HARP(LINE). (f) FHNE.

The process and working principle of the system are shown in Figure 6.

Multilabel classification refers to assigning one or more labels to each sample in a dataset. This section will validate the ability of the FHNE model on the multilabel classification task on the BlogCatalog and PPI datasets. In the previous chapter, we have defined four prediction cases (TP, FP, FN, TN) for binary classification. According to these four cases, this section can calculate the classification precision and recall, where the calculation formula of the accuracy is as follows:

$$Precision = \frac{TP}{TP + FP}. \tag{19}$$

The formula for calculating recall is as follows:

$$Recall = \frac{TP}{TP + FN}. \tag{20}$$

When using precision or recall to evaluate the classification results, if the value of precision or recall is high, it is the most ideal situation. Many times, the values of precision and recall are often not positively correlated. As an extreme example, it is assumed that the number of positive and negative samples in a dataset is 50, the prediction result shows that there is only one positive sample, and the prediction is correct. In this case, the precision result is 100%, but the recall rate is very low, only 1/50. The $F1$ value combines the above two indicators and is a fairer evaluation indicator. The calculation method is as follows:

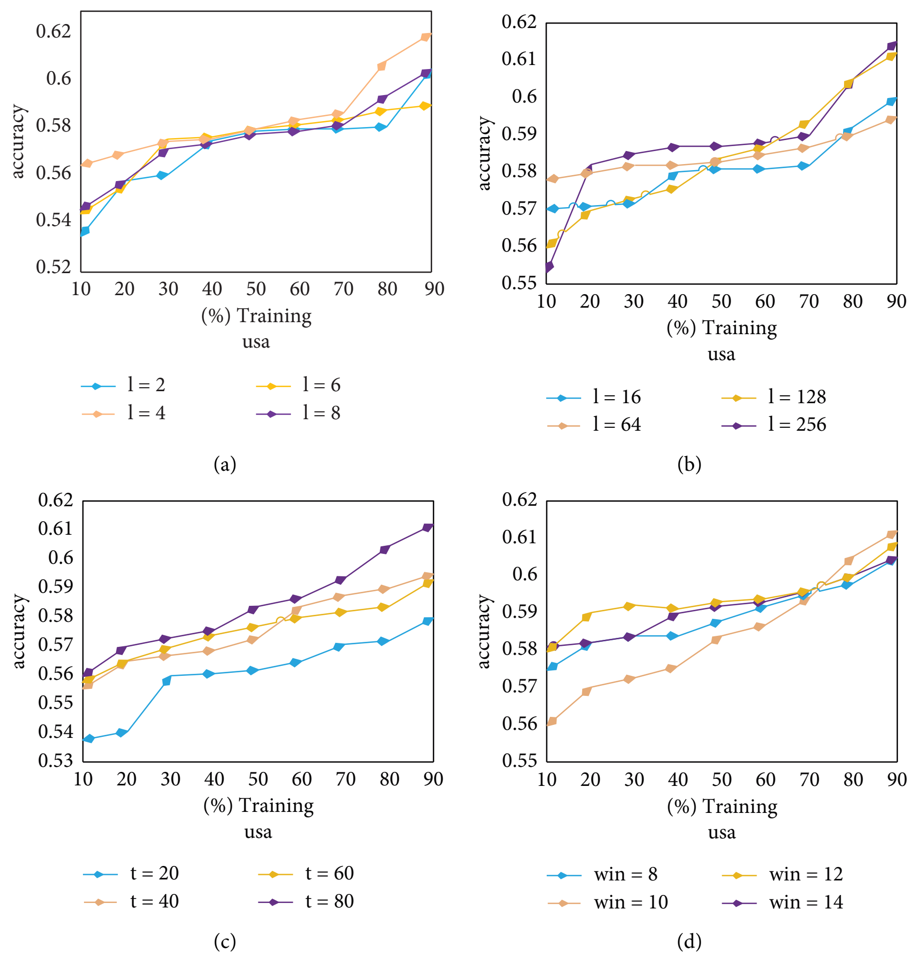


Figure 8: Parameter sensitivity analysis.

$$F1 = \frac{2 * precision * recall}{precision + recall}. \tag{21}$$

This section takes the teaching process of students as an example to verify the ability of the FHNE model in teaching visualization. Since this dataset is relatively small, we directly use the baseline method and FHNE to learn the 2D vector representation of nodes. The visualization results are shown in Figure 7.

As shown in Figures 7(a), 7(b), and 7(e), DeepWalk, node2vec, and HARP(LINE) fail to capture the homogeneity of the network. As shown in Figure 7(c), SDNE can capture the structure of nodes to a certain extent. Most of the light blue nodes are almost coincident (lower left corner), but other types of nodes are scattered. Struc2vec and FHNE capture structural equivalence very well. For the struc2vec algorithm, in Figure 7(d), nodes of the same type have a certain degree of aggregation, but the degree of discrimination of nodes of different types is not large. Overall, FHNE performs better than struc2vec. From Figure 7(f), it can be seen that most of the nodes of the same type are clearly clustered into one class, and the underlying representations between different types of nodes have obvious differences.

The parameter sensitivity analysis of the FHNE model is performed in this work. The FHNE method includes various parameters, and this part confirms, as depicted in Figure 8, how different settings affect the classification performance of student dynamic data. Except for the parameter to be tested, all other parameters have default values.

The classification works best when $l = 4$, because the nodes in the dataset are labeled with four types of labels (corresponding to the number of layers $l = 4$) according to the activity level.

On the basis of the above research, the effect of dynamic student data management is verified, and the effect of resource optimization and student data management is studied, as shown in Tables 1 and 2, respectively.

Table 1: Resource optimization effect.

Number	Resource management	Number	Resource management
1	83.850	13	79.425
2	80.354	14	79.579
3	85.072	15	80.570
4	81.162	16	82.230
5	84.654	17	84.032
6	81.645	18	81.272
7	83.604	19	83.049
8	79.438	20	80.567
9	79.128	21	80.278
10	79.799	22	83.917
11	84.077	23	83.357
12	79.381	24	84.758

Table 2: The effect of student data management.

Number	Student management	Number	Student management
1	86.016	13	92.513
2	92.368	14	84.467
3	89.911	15	87.851
4	87.289	16	92.173
5	89.708	17	91.524
6	84.609	18	90.388
7	91.801	19	89.754
8	85.291	20	92.117
9	85.857	21	84.061
10	89.846	22	88.389
11	91.591	23	84.809
12	89.468	24	87.343

From the above research, we can see that the dynamic student data management using resource optimization technology in the higher education platform proposed in this study can effectively improve the efficiency of college student data management.

4. Conclusion

The student model is one of the core components to realize intelligence, and ITS can implement personalized teaching for learners according to the student model. The student model is a data structure representing the cognitive characteristics of the learner. On the one hand, it captures fundamental data like the student's name, gender, and student ID. On the other hand, it correctly reflects the learner's degree of knowledge, capacity for learning, state of mind, and other factors. The student knowledge base describes and records the student's learning progress and level, as well as the student's error type at each knowledge point. This study combines the intelligent resource optimization technology to construct a dynamic student data management model to improve the management effectiveness of the student data management data model. The research shows that dynamic student data management using resource optimization technology in the higher education platform proposed in this study can effectively improve the efficiency of college student data management.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] J. Holloway, "Teacher evaluation as an onto-epistemic framework," *British Journal of Sociology of Education*, vol. 40, no. 2, pp. 174–189, 2019.

[2] M. Han, N. Zhou, and Z. Qiao, "Analysis of Extension strategies for improving Adaptive teaching ability of college Teachers," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, no. 15, pp. 147–161, 2020.

[3] M. D. C. Salazar, "Interrogating teacher evaluation: Unveiling whiteness as the normative center and moving the margins," *Journal of Teacher Education*, vol. 69, no. 5, pp. 463–476, 2018.

[4] M. L. Derrington and J. W. Campbell, "High-stakes teacher evaluation policy: US principals' perspectives and variations in practice," *Teachers and Teaching*, vol. 24, no. 3, pp. 246–262, 2018.

[5] S. J. Zepeda and A. M. Jimenez, "Teacher Observation and Reliability: Additional Insights Gathered from inter-rater Reliability Analyses," *Journal of Educational Supervision*, vol. 2, no. 2, pp. 11–26, 2019.

[6] E. Brunner, J. M. Cowen, K. O. Strunk, and S. Drake, "Teacher labor market responses to statewide reform: evidence from Michigan," *Educational Evaluation and Policy Analysis*, vol. 41, no. 4, pp. 403–425, 2019.

[7] M. L. Donaldson and S. Woulfin, "From tinkering to going "rogue": how principals use agency when enacting new teacher evaluation systems," *Educational Evaluation and Policy Analysis*, vol. 40, no. 4, pp. 531–556, 2018.

[8] R. Vagi, M. Pivovarova, and W. Miedel Barnard, "Keeping our best? A survival analysis examining a measure of preservice teacher quality and teacher attrition," *Journal of Teacher Education*, vol. 70, no. 2, pp. 115–127, 2019.

[9] K. C. Bastian, D. Lys, and Y. Pan, "A framework for improvement: analyzing performance-assessment scores for evidence-based teacher preparation program reforms," *Journal of Teacher Education*, vol. 69, no. 5, pp. 448–462, 2018.

[10] G. Skedsmo and S. G. Huber, "Top-down and bottom-up approaches to improve educational quality: their intended and unintended consequences," *Educational Assessment, Evaluation and Accountability*, vol. 31, no. 1, pp. 1–4, 2019.

[11] J. A. Grissom, R. S. L. Blissett, and H. Mitani, "Evaluating school principals: Supervisor ratings of principal practice and principal job performance," *Educational Evaluation and Policy Analysis*, vol. 40, no. 3, pp. 446–472, 2018.

[12] K. Kim and Y. Park, "A development and application of the teaching and learning model of artificial intelligence education for elementary students," *Journal of The Korean Association of Information Education*, vol. 21, no. 1, pp. 137–147, 2017.

[13] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education–where are the educators?" *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, pp. 39–27, 2019.

[14] S. C. H. Yang, W. K. Vong, R. B. Sojitra, T. Folke, and P. Shafto, "Mitigating belief projection in explainable artificial intelligence via Bayesian teaching," *Scientific Reports*, vol. 11, no. 1, pp. 9863–9917, 2021.
[15] Y. Lee, "An analysis of the influence of block-type programming language-based artificial intelligence education on the learner's attitude in artificial intelligence," *Journal of The Korean Association of Information Education*, vol. 23, no. 2, pp. 189–196, 2019.
[16] J. M. Alonso, "Teaching explainable artificial intelligence to high school students," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 974–987, 2020.
[17] R. Yang, "Artificial intelligence-based strategies for improving the teaching effect of art Major courses in colleges," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, no. 22, pp. 146–160, 2020.