ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Optimization Strategy of English Education Resource Sharing Platform Based on Cloud Computing Architecture

Xiaolin Wu[1,*]

[1] School of Foreign Languages, Shaoxing University, Shaoxing 312000, Zhejiang, China

SUMMARY: *The current English education resource sharing platform has fragmented resource organization, insufficient granularity in user behavior modeling, and weak responsiveness of the scheduling feedback mechanism, resulting in limited resource aggregation accuracy, targeted recommendation ranking, and system distribution stability. Based on cloud computing architecture, this paper proposes a three-layer optimization strategy that integrates nested semantic tensor modeling, path attention sorting, and edge feedback scheduling control. This paper first uses multimodal tensors to semantically map and restructure resource content to unify resource expression; secondly, combines user behavior vectors with task path attention mechanism to construct a priority recommendation sequence to enhance the consistency of individual recommendations; finally, introduces load-aware index graph and minimum rescheduling cost function to realize dynamic resource path scheduling based on state feedback. Experiments show that the average response time of static load-aware scheduling is 229ms when the number of concurrent requests is 400, the resource matching accuracy of the fusion optimization model is 94.0%, and the node load variance under peak impact of the load level of this strategy is 30.1. This strategy achieves a systematic improvement in resource organization, service recommendation, and scheduling control based on the cloud computing architecture, enhances the platform's sharing efficiency and operational stability, and provides solid support for the efficient integration of lifelong learning and intelligent English education resources.*

KEYWORDS: *Cloud Computing Architecture; English Education Resources; Semantic Tensor Modeling; Personalized Recommendation; Resource Scheduling*

1 Introduction

Under the big background of the fast development of information technology, cloud computing, which has good expandability, strong concurrent handling ability, and flexible resource arrangement, has hence become an important support platform for pushing forward the development of educational informationization. Along with the ceaseless enlargement of the education domain's requirement for resource digital transformation, intellectualized management and on-demand service reaction abilities, English education resource sharing platforms which are constructed upon cloud computing framework have step by step have become one key direction for research and practice work [1, 2]. Due to the diversity, fragmentation and multimodal characteristics of English education resources, higher requirements are placed on the platform's architectural stability, resource scheduling capabilities and intelligent recommendation mechanisms during the integration, classification,

*greece1981@usx.edu.cn
https://doi.org/10.65102/is2026343
Published: 30 April, 2026

retrieval and distribution processes [3, 4]. Existing research uses cloud platform technology to build resource library systems, integrate teaching management function modules, and deploy multimedia resource integration mechanisms. Although phased progress has been made in resource storage efficiency and system response performance, there are still problems such as chaotic resource organization structure, weak semantic association, coarse-grained user portraits, and a single scheduling mechanism [5, 6]. The lack of an effective semantic mapping mechanism between resources leads to duplication and redundancy of similar content within the platform, making it difficult to support semantic-oriented aggregation and push [7, 8]; the lack of high-dimensional modeling and sequence feature extraction of user behavior data leads to low accuracy and consistency in personalized recommendations [9, 10]; the resource path arranging procedure cannot completely perceive the variations of system node condition and resource allocation, hence it brings about obvious undulations of response delay and non-uniform node load in high-concurrency situations [11, 12]. These technical shortcomings manifest themselves during platform operation as decreased resource access efficiency, insufficient relevance of service push, and limited system operation stability. Although some studies have conducted local optimization from the perspectives of edge computing deployment, distributed architecture design, AI-assisted recommendation, and task scheduling algorithms[13, 14], current strategies are mostly based on single-point technology breakthroughs and lack an optimization system with multi-layer linkage and system integration. In the face of high concurrent access to educational resources, multi-source heterogeneous content processing, and personalized service needs, the operating mechanism of the overall platform has not yet formed a closed-loop collaboration, and it is urgent to carry out overall optimization from the underlying architecture, data model, and feedback mechanism [15, 16].

With the support of cloud computing architecture, domestic and foreign scholars have conducted many useful explorations on the construction of English education resource sharing platforms. Zhang[17] designed a teaching resource library system based on a cloud platform, which provides a feasible solution for sharing and promoting high-quality teaching resources through unified data and resource management. His research starts from the perspective of resource integration and emphasizes the importance of the platform in resource standardization and unified management. In order to further enhance the diversity of system functions, Jin[18] designed and constructed an English teaching resource management system that integrates teacher, student and management functions, and significantly improved resource integration capabilities and storage efficiency through cloud computing technology. At the same time, He[19] developed a cloud storage-driven English multimedia teaching resource integration system and combined it with simulation experiments to verify the application potential of the platform in intelligent resource management and dynamic allocation. In order to achieve a more comprehensive system construction goal, Wang and Li[20] systematically built an integrated and efficient English digital resource library platform from the perspective of functional modules, technical architecture to platform deployment process, fully demonstrating the overall supporting role of cloud computing in the development of educational resource platforms. Although the above research has achieved positive results in platform function design and cloud technology application, there are still obvious deficiencies in resource structure optimization, intelligent user modeling, and resource scheduling mechanisms, making it difficult to fully meet the complex and diverse needs of educational resource sharing [21, 22].

With the support of cloud computing, the system optimization strategy of educational resource platforms has gradually become a research hotspot, and the exploration of response efficiency, system stability and intelligent resource allocation has continued to deepen [23,

24]. Mohiuddin et al. [25] improved the response efficiency and stability of the cloud platform in English education scenarios based on edge computing and distributed architecture design, and provided a technical path to meet the platform's high concurrent access and real-time service requirements. In the specific system construction practice, Miao[26] built a university academic affairs management platform based on the cloud service computing system, using distributed storage and management technology to optimize resource utilization. Actual measurements showed that the platform achieved significant improvements in response speed and operating efficiency. In order to gain a deeper understanding of the key aspects of optimization technology, Devi et al. [27] systematically reviewed the mainstream load balancing and task scheduling strategies in cloud computing environments in recent years, focusing on analyzing the role of optimization algorithms and machine learning methods in improving system response efficiency, resource allocation rationality, and platform stability, providing technical reserves for optimization strategies. In addition, Zhang[28] explored the integrated application of cloud computing and artificial intelligence in the education system from a fusion perspective, verified its optimization potential in platform scalability, operational efficiency, and personalized teaching service support, and further expanded the path of intelligent evolution of educational resource systems. Although the above research has provided important support for improving system operation efficiency and resource management capabilities, it still has significant deficiencies in multi-strategy collaborative optimization, adaptive scheduling in complex environments, and semantic matching of educational resources, making it difficult to meet the deep needs of educational platforms for high-precision and intelligent optimization strategies [29, 30].

According to the problems above, this paper puts forward a three-layer system optimization strategy which integrates resource modeling, intelligent arrangement, and dynamic arrangement for an English education resource sharing platform under the cloud computing structure. By working together from three dimensions: bottom-level resource semantic structure reconstruction, mid-level user behavior path modeling, and upper-layer scheduling feedback control, it systematically promotes the platform's resource organization efficiency, recommendation precision degree, and service response stable performance. In terms of resource modeling, this paper constructs a multimodal nested tensor model to uniformly encode teaching resources, introduces semantic embedding mapping and content structure reconstruction mechanism to eliminate resource heterogeneity, establish a high-dimensional content association network, and optimize resource storage structure and query efficiency. In terms of user modeling and recommendation ranking, this paper combines the user's historical behavior sequence with the platform task node path to build a path attention mechanism, integrating the attention-weighted behavior vector with the sequence sparse representation to improve the ranking stability and recommendation consistency. In terms of resource scheduling mechanism design, this paper introduces an edge feedback scheduling mechanism and a load-aware index graph, constructs a minimum rescheduling cost function based on the platform's operating status, adjusts and allocates resource paths in real time, and enhances system scheduling elasticity and service response agility in high-concurrency environments. The three types of strategies are based on unified resource representation, mediated by behavior sequence sorting, and controlled by state-aware scheduling, forming a three-layer linkage optimization path from resource organization, service push to node scheduling, and constructing an intelligent management and efficient service platform architecture for complex English teaching resources. Compared with the existing literature that focuses on edge deployment, functional integration or single scheduling optimization, this method emphasizes cross-level data expression consistency and feedback scheduling closed-loop design, effectively responding to the shortcomings of

previous research in structural integration, path optimization and intelligent allocation. Based on existing research results, the method proposed in this paper starts from the perspective of the entire platform process and systematically reconstructs the resource expression method, behavior modeling mechanism and scheduling feedback path. It expands the comprehensive optimization model of the educational resource cloud platform from data organization to service response, and provides technical support and design ideas for building a stable, efficient and sustainable English education resource sharing platform.

2 English Education Resource Optimization Strategies in a Cloud Computing Architecture

2.1 Multimodal Tensor Modeling and Resource Semantic Reconstruction Based on a Cloud-Native Graph Database

Under the cloud computing architecture, in response to the high heterogeneity of English education resources in content structure and semantic expression, relying on the distributed storage and efficient relationship processing capabilities of cloud-native graph databases, this paper adopts a multimodal tensor modeling method to perform high-dimensional expression and semantic unification of resource data, aiming to improve the accuracy of resource matching and the data organization efficiency of the platform. This method takes the text, audio and video modal data of the resource as input, constructs a third-order tensor representation, and completes semantic reconstruction through a nested semantic mapping mechanism.

Supposing the original resource set is $R = \{r_1, r_2, ..., r_N\}$, and the multimodal input corresponding to each resource r_i is a triplet (x_i^t, x_i^a, x_i^v), where $x_i^t \in R^{d_t}$ represents the text modality feature, $x_i^a \in R^{d_a}$ represents the audio modality feature, and $x_i^v \in R^{d_v}$ represents the video modality feature. A multimodal encoder is used to map it to a unified semantic space to obtain a fusion vector $x_i = f_\theta(x_i^t, x_i^a, x_i^v) \in R^d$, where f_θ is a parameter-trainable fusion network. The fusion method uses a gated attention mechanism to adjust the modality weights, which is defined as:

$$x_i = \sigma(W_g\,[x_i^t; x_i^a; x_i^v] + b_g) \odot tanh(w_f\,[x_i^t; x_i^a; x_i^v] + b_f) \tag{1}$$

In formula (1), W_g , $w_f \in R^{d\times(d_t+d_a+d_v)}$ is the weight matrix, $b_g, b_f \in R^d$ is the bias term, $\sigma(\cdot)$ is the Sigmoid activation function, and $\odot$ represents the Hadamard product. This mechanism enhances the dynamic selection capability of inter-modal information and achieves deep fusion of semantic features.

The fused resource representation vectors are persistently stored in the cloud-native graph database and form a third-order tensor $X \in R^{N\times d\times T}$, where T is the number of semantic channels. A high-level semantic structure is reconstructed for X, capturing potential semantic relationships through tensor decomposition. Tucker decomposition is used:

$$X \approx G \times_1 U_1 \times_2 U_2 \times_3 U_3 \tag{2}$$

In formula (2), $G \in R^{r1\times r2\times r3}$ is the core tensor, $U_1 \in R^{N\times r1}$, $U_2 \in R^{d\times r2}$ and $U_3 \in R^{Txr3}$ are factor matrices, and $\times_n$ represents the n-th modular multiplication operation. This decomposition achieves dimensionality reduction and reconstruction of resource semantic features, effectively removing modal redundancy and semantic complexity. The graph

structure of a graph database is naturally suitable for storing and efficiently querying the complex semantic relationship network between resources implied by the factor matrix obtained by decomposition. At the same time, the distributed storage capabilities of the cloud platform ensure reliable storage and fast access to large-scale tensor data.

In order to unify the resource expression, a semantic alignment loss function is introduced to supervise the training of the reconstructed tensor features. The reconstructed semantic embedding is defined as z_i, which should maintain the minimum semantic distance with the standard semantic label embedding y_i. The loss function is:

$$L_{sem}=\frac{1}{N}\sum_{i=1}^{N} // z_i-y_i //^2 \tag{3}$$

In formula (3), $z_i = U_1^t x_i, y_i$ is generated by the teacher model or given by the expert label embedding model. This mechanism strengthens the semantic consistency of resource representation in a unified semantic space and improves the semantic matching accuracy in subsequent tasks. Graph databases support efficient nearest neighbor search for embedded vectors and their associations, providing underlying support for semantic alignment and subsequent matching.

During model training, the Adam optimizer is used to minimize L_{sem} as the objective, and the tensor decomposition and reconstruction error L_{rec} is combined to form a joint loss:

$$\widetilde{L_{total}} =L_{sem} + \lambda \cdot L_{rec} , L_{rec} = \|X - \hat{X}\|^2\_F \|X - X^2\|_F \tag{4}$$

In formula (4), λ is the weight coefficient, and $\|\cdot\|_F$ stands for the Frobenius norm. After the training work gets finished, every item of resource data is mapped into a unified semantic tensor space, and feature compression and semantic enhancement are realized by means of structural reconstruction. The ultimate resource semantic graph and its tensor expression are effectively stored in a distributed cloud-origin graph database, hence supporting the platform's exact matching and effective arrangement of multi-modal resources. The cloud platform's flexible calculation resources guarantee the efficiency of large-scale tensor calculation operations and graph traversal searches, while the dispersed storage structure offers a firm base for the long-term storage and high-concurrency visiting of huge different kinds of educational resources.

2.2 User Behavior Modeling and Path-Attention Ranking Mechanism

With the support of cloud computing architecture, in order to improve the stability and personalized adaptation of English education resource recommendation results, this paper extracts user task intentions and generates a consistent and optimized resource recommendation sequence by constructing a path modeling mechanism and attention ranking model based on user behavior trajectories. This method takes the user's operation sequence and context state on the platform as input, constructs a state transition graph and integrates the path attention mechanism to achieve recommendation sorting.

Let the user set be $U = \{u_1, u_2, ..., u_M\}$, and the behavior sequence corresponding to each user u_i be recorded as $S_i = \{a_i^1, a_i^2, ..., a_i^t\}$, where a_i^t represents the t-th behavior operation, and the operation types include resource browsing, collection, completion rate feedback, etc. The behavior sequence is state-embedded, and the behavior type encoding vector $e(a_i^t) \in R^{d_a}$ and the context feature $c_i^t \in R^d$ are jointly input into the behavior encoder to obtain the state representation:

$$h_i^t = \tanh(W_a \cdot e(a_i^t) + w_c \cdot c_i^t + b) \tag{5}$$

In formula (5), $W_a \in R^{d_h \times d_a}$, $w_c \in R^{d_h \times d}$, $b \in R^{d_h}$ and $h_i^t \in R^{d_h}$ are the state vectors of the user at time t. S_i is mapped to the state path graph $G_i = (V_i, E_i)$, where the node set V_i corresponds to the user behavior state and the edge set E_i represents the temporal transfer dependency.

The attention mechanism on path level is then utilized on the state graph, thus to give different weight values to candidate behavior paths. Paths which can better reflect the user's present learning objective make greater contributions to the aggregated preference vector, hence actions that are isolated or have weak relations are given lower weights. By this method, the ranking model that we study is pushed by structural behavior data, hence not only by original click frequency itself.

$$\alpha_i^k = \text{softmax}(q^t \cdot \tanh(W_p \cdot \bar{h}_i^k + b_p)) \tag{6}$$

In formula (6), $\bar{h}_i^k = (1/l) \sum_j h_i^{t_j}$ is the average representation of path p_i^k, q^t and W_p are the parameters of the path attention mechanism. The final behavior representation is obtained by weighted aggregation path embedding:

$$s_i = \sum_k \alpha_i^k \cdot \bar{h}_i^{-k} \tag{7}$$

In formula (7), the behavior representation s_i represents the user's current task preference. The resource ranking score function is constructed, and the resource semantic vector x_j and s_i are input into the scoring function:

$$r_{ij} = s_i^t \cdot W_r \cdot x_j + b_r \tag{8}$$

In formula (8), $W_r \in R^{d \times d_h}$ and $b_r \in R$ are linear scoring parameters. Based on r_{ij}, the candidate resources are sorted in descending order to generate the recommendation sequence $R_i = \text{sort}(\{r_{ij}\})$.

To enhance the consistency of recommendation results, a consistency loss function is defined and sorted, with the recommendation deviation in multiple rounds of behavior sequences as the penalty term. Let the previous round of recommendation sequence be R_i^{t-1} and the current one be R_i^t, and define the consistency loss as:

$$L_{\text{sort}} = \frac{1}{K} \sum_i \pi_i^{t_j} - \pi_i^{t_{j-1}} \tag{9}$$

In formula (9), $\pi_i^{t_j}$ represents the position index of resource j in the recommendation sequence R_i^t. The recommendation ranking error and consistency loss are jointly optimized to construct the total loss function:

$$\widehat{L_{\text{total}}} = L_{\text{rec}} + \beta \cdot L_{\text{sort}} \tag{10}$$

In formula (10), β is a weight hyperparameter. Backpropagation updates all parameters using the momentum-based Adamw optimizer, ultimately implementing a dynamic ranking mechanism based on user task path awareness and generating personalized recommendation sequences with semantic matching and behavioral consistency.

2.3 Edge Feedback Scheduling Control and Dynamic Resource Scheduling

In the cloud computing architecture, to improve the service responsiveness and distribution stability of the English education resource sharing platform in a high-concurrency environment, this paper constructs a dynamic scheduling mechanism based on edge node feedback, integrates the state-aware control model with the rescheduling cost function, and implements a dynamic resource path distribution strategy for load balancing and minimal reconfiguration overhead. This mechanism builds a scheduling graph model based on resource request status, node load status and network topology relationship, and implements multi-objective optimal path search and feedback-driven real-time scheduling adjustment.

Defining the platform edge computing node set as $N = \{n_1, n_2, ..., n_j\}$ and the resource request set as $Q = \{q_1, q_2, ..., q_T\}$. Each request $q_t \in Q$ is initiated by the user and contains a resource type vector $r_t \in R^{dr}$, a target delay threshold τ_t, and a QoS requirement vector $\eta_t \in R^d\eta$. The states of all edge nodes are modeled as a dynamic state vector $s_j^t = [\lambda_j^t, \mu_j^t, \rho_j^t]$, where λ_j^t represents the number of requests arriving at node n_j per unit time, μ_j^t is the maximum processing capacity of the node per unit time, and $\rho_j^t = \lambda_j^t / \mu_j^t$ represents the instantaneous load rate of the node. The load-aware scheduling graph is modeled as a directed graph $G_t = (N, E_t)$. The edge set E_t represents the resource accessibility relationship between nodes, and the edge weight is the response cost function:

$$C_t(i, j) = \alpha \cdot d_{ij} + \beta \cdot \rho_j^t + \gamma \cdot l_j^t \tag{11}$$

In formula (11), d_{ij} represents the network delay from node i to j, ρ_j^t is the current load rate of node j, l_j^t is the average queue length on node j, and α, β, and γ are normalized weight coefficients that satisfy $\alpha + \beta + \gamma = 1$. Based on the target node path search for resource request $q_t$by G_t, the improved Dijkstra algorithm with feedback control is adopted to dynamically adjust the candidate path cost according to the edge weight in each iteration.

After the resource scheduling path is determined, a state feedback mapping function $\Phi(\cdot)$ is constructed for the current scheduling state and the historical scheduling records to guide whether to trigger the rescheduling operation. Defining the scheduling state of the current request on node n_j as $\delta_{tj} \in \{0,1\}$, and 1 means it has been distributed to the node. The rescheduling decision is based on the cost function $\Delta C(q_t, n_j \rightarrow n_k)$:

$$\Delta C(q_t, n_j \rightarrow n_k) = C_t(j, k) + \theta \cdot \Omega(q_t, n_j) \tag{12}$$

In formula (12), $\Omega(q_t, n_j)$ represents the interruption cost caused by migrating request q_t from node n_j to n_k, θ is the penalty factor, and $C_t(j, k)$ represents the response delay adjustment value caused by the path change. If $\Delta C(q_t, n_j \rightarrow n_k) < \varepsilon$, the system performs a migration rescheduling operation, and ε is the fault tolerance threshold dynamically set by the system. The calculation of $\Omega(q_t, n_j)$ depends on the resource utilization rate and processing time of the migration request, expressed as:

$$\Omega(q_t, n_j) = \zeta \cdot (t_t^p / t_t^{max}) \cdot \xi(q_t) \tag{13}$$

In formula (13), t_t^p is the running time of $q_t$on node n_j, t_t^{max} is its maximum tolerated processing cycle, $\xi(q_t)$ represents the resource demand weight function, and ζ is the migration penalty proportional constant. Combined with the dynamic feedback control mechanism of ΔC, the scheduling decision function is executed on all requests within the system scheduling period T:

$$\delta_{tk} = argmin_{nk} \in N\,[C_t(j, k) + \theta \cdot \Omega(q_t, n_j)] \tag{14}$$

In formula (14), this mechanism realizes the optimal path switching with the minimum reconfiguration cost based on the current scheduling state, thereby maintaining low latency and high stability of the system service chain.

The overall scheduling efficiency of the system is evaluated by the average response time index $\bar{\tau}$ and the node load variance $\sigma^2(\rho)$, where:

$$\bar{t}=\frac{1}{T}\sum_t\ t_\tau \tag{15}$$

In formula (15), t_τ is the actual response time of request q_t.

$$\sigma^2(\rho)=\frac{1}{J}\sum_i\ (\rho_i^t-\bar{\rho})^2 \tag{16}$$

In formula (16), $\bar{\rho}$ is the average node load rate.

The feedback scheduling control strategy not only ensures the accuracy of resource allocation, but also reduces queue congestion and system load fluctuations in high-concurrency scenarios, and achieves dynamic balance of resource distribution among edge computing nodes and optimization of service stability.

2.4 Comprehensive Implementation and System Design of the Three-Layer Optimization Strategy

The systematic integration of the three-layer optimization strategy is based on the cloud computing resource virtualization environment, building a multi-module collaborative semantically driven recommendation and scheduling system. In this system, the semantic embedding of resources, user path preference generation and edge load scheduling process form an end-to-end dynamic feedback loop through shared representation space and linkage state flow.

The high-level representation $Z = \{z_1, z_2, \ldots, z_N\}$ of resource information after tensor modeling is encoded into the shared representation matrix $H_r \in R^{N\times d}$, which together with the user state space $H_u \in R^{M\times d}$ constructs the system semantic field: $F_ss=H_rH_u^T$. In this semantic field, the interaction potential between any resource z_i and user u_j is defined by bilinear tensor projection as:

$$\Psi_{ij}=\langle z_i,T,u_j\rangle=\sum_{a=1}^{d}\ \sum_{b=1}^{d}\ \sum_{c=1}^{d}\ z_i^{(a)}\cdot T_{abc}\cdot u_j^{(b)} \tag{17}$$

In formula (17), T denotes the third-order fusion kernel tensor of the system, and Ψij represents the semantic adaptability of the current resource-user pair in the global state field. This term is used to connect resource semantics with collaborative behavior generation during training.

Based on the above Ψ_{ij}, the system constructs a joint sorting and distribution priority function:

$$\Gamma_{ij}(t)=\sigma\big(\Psi_{ij}+\varphi(u_j,t)-\kappa(z_i,t)\big) \tag{18}$$

In formula (18), φ(uj,t) denotes the historical behavior density adjustment item of user uj within the current window, and κ(zi,t) denotes the update tension of resource zi on the edge node. Γij(t) is therefore interpreted as the dynamic distribution score of a user-resource pair

and is used to drive node selection.

The overall service scheduling function of the system is determined by the following formula:

$$\delta_t(i,j)=\arg\max_{k\in N}\left\{\Gamma_{ij}(t)\cdot(1-\chi_k(t))-\lambda\cdot D(i,k)\right\} \tag{19}$$

In formula (19), Xk(t) is the normalized load rate of node k at time t, D(i,k) is the transmission cost from resource i to node k, and λ is the cost penalty coefficient. The function integrates semantic suitability, user activity intensity, and runtime load into a unified allocation decision.

To achieve joint parameter learning and performance synergy across the entire system, a joint optimization objective is defined:

$$\min_{\theta_r,\theta_u,T} J=E_{(i,j,t)}\left[l\left(y_{ij},\Gamma_{ij}(t)\right)+\eta_1\cdot /\!/ \nabla T /\!/ _F^2+\eta_2\cdot KL(P_\delta /\!/ Q_\delta^*)\right] \tag{20}$$

In formula (20), $\ell(\cdot)$ denotes the binary cross-entropy term for recommendation matching, $\|\nabla T\|F^2$ controls the complexity of the semantic tensor core, and $KL(P\delta\|Q*\delta)$ measures the deviation between the actual scheduling distribution and the expected stable distribution. Joint optimization of these terms enables the three-layer model to balance accuracy, compactness, and scheduling stability.

At deployment, the semantic modeling module outputs the tensor core and resource embeddings, the behavior module outputs user state vectors and density estimates, and the scheduling controller reads real-time node load to execute distribution decisions. Kafka is used as the asynchronous coordination layer so that state updates and parameter broadcasts can be propagated without blocking online service requests.

The resulting architecture forms a closed loop from resource representation to recommendation and then to resource distribution. Figure 1 summarizes the information flow and module dependencies of this three-layer design.

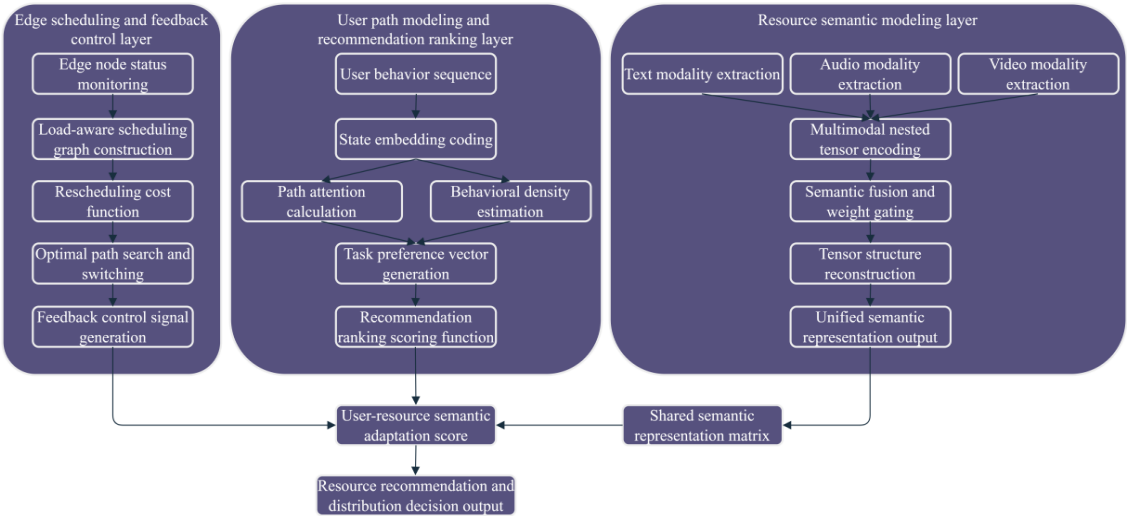

Figure 1: System optimization structure integrating modeling, recommendation and scheduling three-layer mechanism

Figure 1 takes the resource semantic modeling layer, user path modeling and recommendation ranking layer, and edge scheduling and feedback control layer as three main horizontal structures, and exhibits the processing logic and information flow approaches of

key processes including multimodal input and semantic unification, behavior trajectory extraction and path-aware recommendation, state monitoring and cost-constrained scheduling on each level. The semantic modeling level makes a shared semantic expression via mode taking-out, tensor combination and structure rebuilding, hence it gives an organized semantic base for following behavior modeling; the user modeling level builds state coding, density calculating and path focus models, and therefore outputs recommendation trend vectors under dynamic task perception; the arrangement controlling layer takes real-time node condition as input to push the cost function and route changing mechanism to finish dynamic allocation. The output results of these three layers gather together to form a semantically adjusted and common expression matrix, thus pushing the final recommendation choices and resource distribution route production. This architecture reinforces the connection mechanism among semantic consistency, behavioral correctness and scheduling stability, hence it realizes the system-level optimization from resource expression to service allocation.

3 Experimental Evaluation of Optimization Strategies in a Cloud Computing Platform

3.1 Comprehensive Analysis of English Education Resource Matching Performance Integrating Multimodal Tensors and Structural Optimization

For the purpose of confirming the influence of different optimization methods on the matching precision of English education resources, this study systematically changed the resource organization and semantic processing system of the platform in the distributed environment of Alibaba Cloud ECS (Elastic Compute Service) cluster (1), step by step brought in four technical roads: multimodal modeling, semantic reconstruction, structural optimization, and fusion strategy, and took the original platform as a control to construct a multi-version contrast experiment environment. The experiment data are deposited in the cloud-native graph database Neo4j, and parallel acceleration of tensor operations is realized by the Spark distributed calculating frame. In the aspect of resource dimensions, four kinds of typical teaching resources, containing vocabulary, grammar, reading understanding and writing, are included to test the performance differences of different strategies in handling resources with different semantic complication and structural heterogeneity, therefore to study the connection between the depth of structural modeling and the semantic unification of resources. Figure 2 has displayed the distributing circumstance of matching accurate degree for different resource sorts under every optimization strategy.

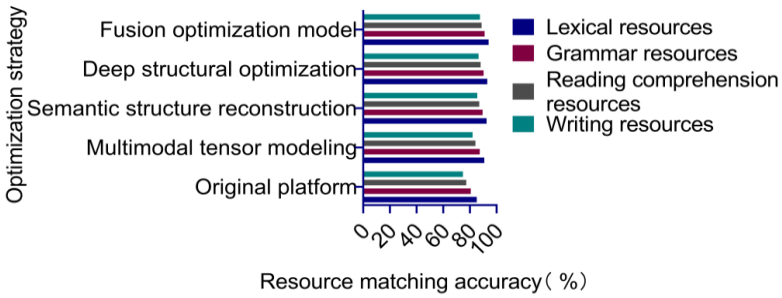

Figure 2: Resource matching accuracy of different optimization strategies in multiple types of English education resources

According to what Figure 2 displays, the fusion optimization model obtains the tallest matching precision for every resource kind, wherein vocabulary resources achieve 94.0%, grammar resources achieve 91.1%, reading comprehension resources achieve 88.8%, and writing resources achieve 87.5%, therefore the total average matching accuracy is 90.3%. This method effectively promotes the platform's semantic analysis and structure matching abilities for different types of content by means of unified semantic tensor space and multi-layer feedback route optimization. By way of comparison, deep structure optimization obtains a little worse results in grammar and writing resources, but the total average matching accuracy still attains 89.5%, which thus displays the core function of structural hierarchy refinement for the promotion of semantic fusion abilities. The whole average matching accuracy degrees of semantic structure rebuilding and multi-mode tensor model building are 88.7% and 86.1%, respectively, hence it shows that even though a single optimization dimension can raise the matching effect of some resources, it still has certain restrictions in dealing with the diversity of tasks. The whole average matching correctness of the original platform is 79.5%, which is especially low in writing and reading comprehension resources, this reflects its not enough semantic recognition and expression consistency abilities. On the whole, the layer-by-layer bringing in of optimization strategies has greatly promoted the accuracy of platform resource matching, hence this fusion model possesses the most top resource adaptability and generalization performance.

3.2 Optimizing Recommendation Consistency with the Path Attention Mechanism

This experiment studies the effect of different recommendation methods on recommending English study resources, it specially aims at four resource kinds: word, grammar, reading understanding, and writing, and assesses the matching correctness of each method in actual use. Through the comparison of recommendation methods including collaborative filtering, content recommendation, sequence modeling, path attention mechanism, as well as path attention and user modeling, this experiment carries out an analysis on how these strategies promote the consistency of recommendations under different types of resources. Figure 3 displays the distributing of consistence marks for every recommend strategy in different resource kinds.

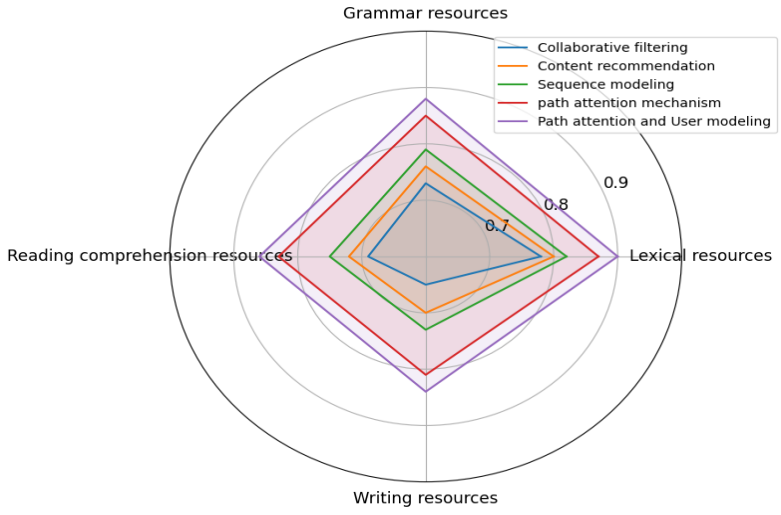

Figure 3: Comparison of recommendation consistency for each resource type under different recommendation strategies

From Figure 3 we can see, the combination strategy of path attention and user modeling obtains the best performance on all resource types, vocabulary resources get 0.9 score, grammar resources get 0.88, reading comprehension resources get 0.86, and writing resources get 0.84. The whole average recommendation consistence score attains 0.87. By comparison, the cooperative filtering method generally obtains low scores on different resources, its recommendation consistency score is only 0.65 for writing resources, which shows this method has limitations when it deals with complicated semantics and various study requirements. The score values of the path attention mechanism and the sequence modeling strategy are relatively close, and their performance is better than that of collaborative filtering and content recommendation on every resource type, hence it indicates that the modeling methods based on path and sequence can better capture the dependence relations between resources and the dynamic changes of user demands.

In addition, this experiment moreover has the purpose to probe the recommendation consistency behavior of various recommendation methods under different user interest complication degrees. Through the comparison of five recommendation methods, which include collaborative filtering, content recommendation, sequence modeling, path attention mechanism, and path attention together with user modeling, their influences are analyzed under different kinds of interest types, for example, single interest users, medium interest diffusion users, high interest diffusion users, multimodal interest users, and unstable preference users. Figure 4 gives out the score distribution of recommendation consistency of the five recommendation methods under different degrees of user interest complexity.

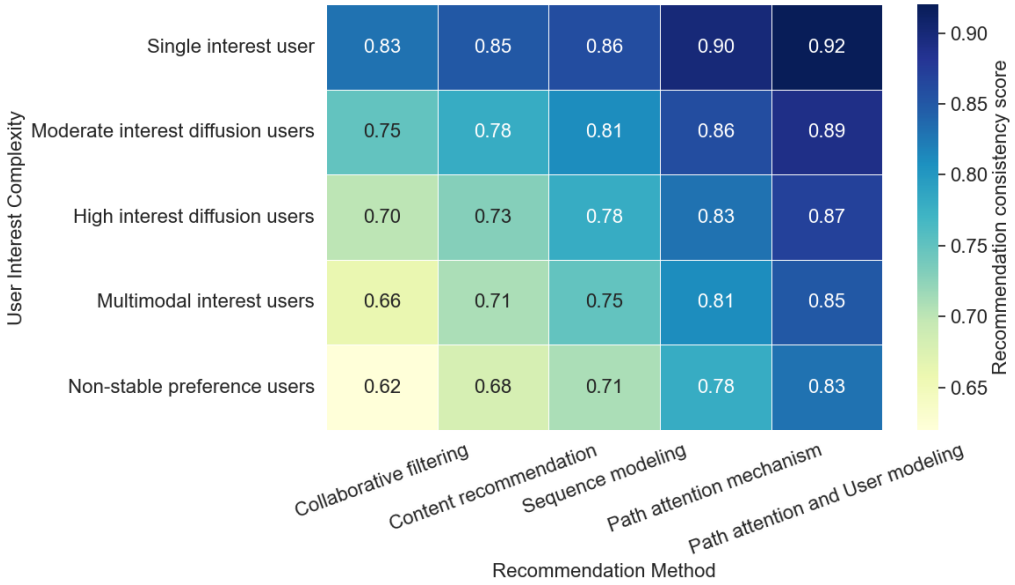

Figure 4: Comparison of recommendation consistency scores between recommendation methods and user interest complexity

From Figure 4 we can see that, the method which combines path attention and user modeling has good performance under all levels of user interest complexity. In the user group that only has single interest, the score reaches as high as 0.92, thus it shows the superior property of this strategy when it does the work of accurately matching the need of users. To users who have medium and high interest diffusion, the scores of the method which combines path attention and user modeling are 0.89 and 0.87 respectively, these scores still keep in a high level, hence it shows that this method possesses strong adaptive ability for user interests. By way of comparison, the collaborative filtering method obtains low evaluation scores

among all user groups. Among users who have preferences that are not stable, the recommendation consistency score is only 0.62, hence it indicates that this score cannot effectively grasp the complex and unstable user interest requirements. The sequence modeling and path attention mechanism also display certain superiorities when handling users who have multimodal interests and unstable preferences, but on the whole they still fall behind the combination of path attention and user modeling. On the whole, the combination of path attention mechanism and user modeling not merely obtains good effect for users who hold a single interest, but also can more well adapt to user groups that have more complicated interests, hence enhancing the accuracy and consistency of recommendation results.

3.3 Load-Aware Scheduling for Response Time Optimization

For assessing the influence that arrangement tactics bring to the service reaction speed of cloud calculation platforms under high concurrency situations, this experiment has designed five representative arrangement methods: circular average arrangement, shortest queue arrangement, weight addition arrangement, static load perception arrangement, and dynamic load perception arrangement. When the count of concurrent requests raises from 50 to 400, the average response time data which correspond to every strategy are collected to analyze the influence that the scheduling algorithm has on the system resource distribution efficiency and load balancing ability. Figure 5 has displayed the response time curves of the five scheduling strategies which are under different quantities of concurrent requests.

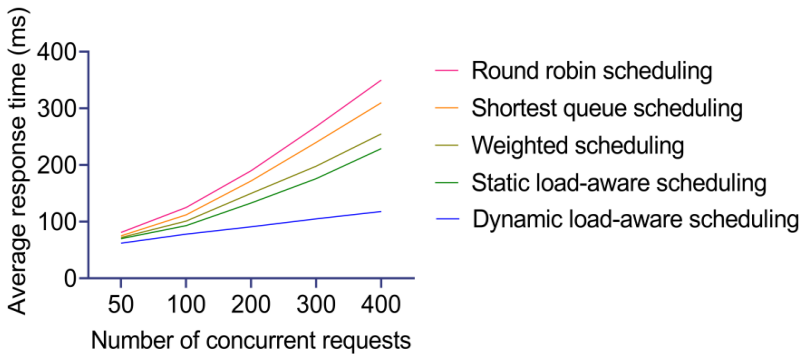

Figure 5: Comparison of average response time as a function of concurrent requests under different scheduling strategies

Figure 5 can tell us that along with the increment of concurrent request quantity, the average response time of every scheduling strategy goes up, but the growth rate has obvious differences. The response speed of polling scheduling has the most obvious increase, rising to 350ms when there are 400 requests, therefore it indicates that in high-concurrency situations it is hard to effectively assign tasks. The shortest queue dispatching and weight-based dispatching perform a little better when load is medium, but they rise to 310ms and 255ms respectively when the quantity of requests gets to 400, hence this shows that they still have resource bottlenecks under the policies of queue length and weight. Scheduling which knows static load can promote allocation efficiency through setting node load weights in advance. Its time of reaction is 229ms in the condition of 400 concurrent requests, thus it displays a certain degree of capability of anti-interference. Dynamic load aware dispatching keeps the smallest response time in the whole process, rising from 62ms to 118ms with very small growth, which shows its advantage in real-time feeling of node load and dynamic changing of methods. This strategy can hold steady service reply when concurrent pressure goes up, and it is a key arranging mechanism for enhancing platform working efficiency.

3.4 Optimization of Node Load Variance by Scheduling Control

For the purpose of assessing the influence of diverse node scheduling schemes upon the equilibrium of platform resource allocation under diverse load degrees, this experiment commences from five degrees from light load to peak impact, and establishes five scheduling approaches: non-optimized scheduling, traditional balanced scheduling, edge feedback scheduling, dynamic migration scheduling, and the optimization strategy that is proposed in the present paper. Through the comparison of node load variance numerical values, the effect of the scheduling scheme on distribution balance can thus be reflected by us. Figure 6 displays the changing tendency of node load variance for each strategy under different load levels.

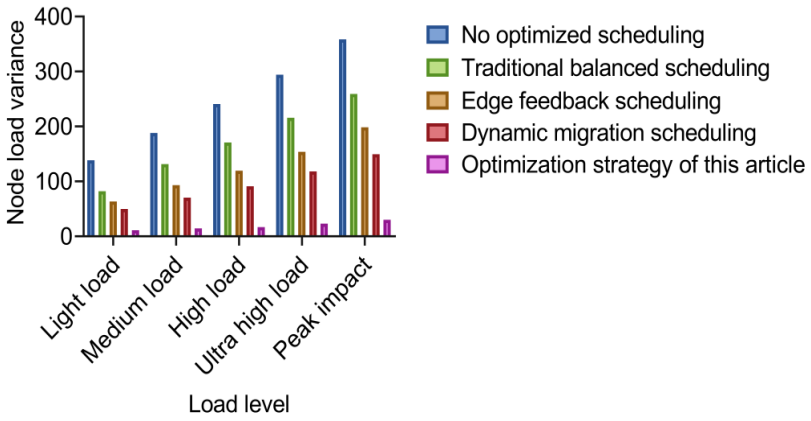

Figure 6: Comparison of node load variance under different scheduling strategies at multiple load levels

As what Figure 6 has shown, the optimization strategy that this paper puts forward displays quite obvious load balancing advantages on every load level. Under the situations of light load and medium load, the variances of node load are 11.2 and 14.6, separately, they are lower than the 49.8 and 70.5 which correspond to the dynamic migration scheduling. This difference mainly comes from the dynamic perception and route adjustment mechanism that is introduced in this strategy, which therefore effectively avoids the early gathering effect of load deviation. In the high-load to ultra-high-load phases, the load variation of the strategy put forward by this paper is 16.9 and 23.3, separately, while for edge feedback scheduling it is 119.4 and 153.8 respectively. The difference has significance, hence it indicates that the strategy which is proposed in this paper possesses stronger allocation flexibility in the stage which is task-intensive. Under peak collision effect, the variance of traditional balanced dispatching achieves 259.2, while the strategy put forward by this paper still keeps at 30.1, hence it reflects that this strategy has dispatching stability and response precision under high-pressure situations. The whole data tendency shows that the optimization method in this article can obviously decrease node load undulations in a multi-layer load environment and hence is an effective method for promoting the whole scheduling efficiency of the platform.

3.5 Synergistic Effects of Optimization Strategies and Improved Overall Platform Performance

For the purpose of comprehensively measuring the influence that different system optimization strategy combinations exert on the key performance indicators of the platform, this experiment chooses three technical roads: tensor modeling A, path attention ranking B,

and load-aware scheduling C, builds seven kinds of strategy combinations, and hence carries out assessments from four dimensions: user satisfaction degree, resource usage rate, successful request ratio, and system load balancing. This paper adopts many groups of comparison experiments to analyze the change tendencies of different indexes under the overlapping of strategies, therefore it reveals that these strategies have the synergistic promotion effect to the whole system performance. Figure 7 has displayed the percent proportion result of every optimization combined strategy on the four index items.

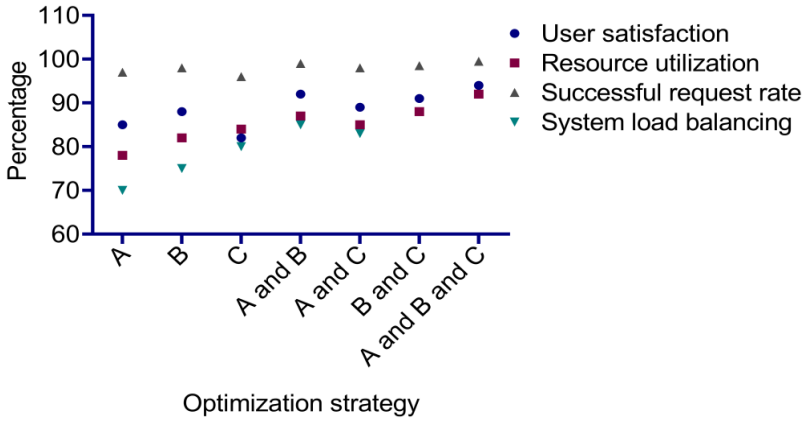

Figure 7: Comparison of key performance indicators of the platform under multi-strategy optimization combination

This paper puts together multimodal tensor modeling, path attention sorting, and edge feedback scheduling control, therefore puts forward an optimization tactic for an English education resource sharing platform that is based on cloud computing architecture, hence the goal is to promote the platform's resource matching accuracy, recommendation consistency, and service response stability. Experiment consequences indicate that the whole resource matching correctness of the fusion optimization model attains 90.3%, the recommendation consistency mark of the combined path attention and user modeling tactic for high-interest diffusion users is 0.87, and the mean reaction time of the dynamic load-aware scheduling tactic is 118ms when the quantity of concurrent requests is 400, hence proving the notable function of this tactic in promoting platform sharing efficiency and stability. Although this strategy has good performance in resource matching and scheduling optimization, hence under high load situations, the scheduling mechanism needs frequent adjustments, which thus may influence the experience in usage of partial users.

4 Conclusion

This current paper puts forward a three-layer optimization framework for cloud-based English education resource sharing which connects multimodal semantic reconstruction, path-attention recommendation, and edge-feedback scheduling. The experimental results display that the fusion model promotes the average accuracy of resource matching to 90.3%, the combined strategy of path attention and user modeling lifts recommendation consistency to 0.87 among different resource types, and dynamic scheduling which considers load restricts response time to 118 ms when concurrent requests are 400, meanwhile it makes the peak variance of node load maintain at 30.1. These outcomes indicate that the method put forward promotes not only single modules but also the cooperation among resource arrangement,

individualized service, and operation distribution. The main restriction is that frequent rearranging of plans may still raise control expense under continuous extreme load, which hence should be solved in future work through adding more explicit forecasting of sudden flow traffic and moving cost.

About the Author

Xiaolin Wu was born in Xinchang, Zhejiang Province, China, in 1981.She is a lecturer in Shaoxing Univeristy. She received the bachelor's degree from Shaoxing University, her master's degree from Shanghai University. Her research interests include language teaching, translation and discourse analysis.

References

[1] Wang, C., & Wang, D. (2023). Managing the integration of teaching resources for college physical education using intelligent edge-cloud computing. Journal of Cloud Computing, 12, 82.

[2] Zhao, L., Hu, G., & Xu, Y. (2024). Educational resource private cloud platform based on OpenStack. Computers, 13(9), 241.

[3] Tahir, S., Hafeez, Y., Abbas, M. A., Nawaz, A., & Hamid, B. (2022). Smart learning objects retrieval for e-learning with contextual recommendation based on collaborative filtering. Education and Information Technologies, 27(6), 8631-8668.

[4] Li, Y., Liang, Y., Yang, R., Qiu, J., Zhang, C., & Zhang, X. (2024). CourseKG: An educational knowledge graph based on course information for precision teaching. Applied Sciences, 14(7), 2710.

[5] Zhang, X. (2022). Cloud storage system of teaching resources based on internet of things. International Journal of Continuing Engineering Education and Life Long Learning, 32(6), 699-713.

[6] Qu, K., Li, K. C., Wong, B. T. M., Wu, M. M. F., & Liu, M. (2024). A survey of knowledge graph approaches and applications in education. Electronics, 13(13), 2537.

[7] Hamdan, N. M., Admodisastro, N., Osman, H. B., & Muhammad, M. S. B. (2024). Semantic interoperability in multi-cloud platforms: A reference architecture utilizing an ontology-based approach. International Journal on Advanced Science, Engineering and Information Technology, 14(6), 1967-1975.

[8] Shen, Y., Yu, G., Liu, X., et al. (2024). Resource sharing and allocation excitation mechanism of teaching cloud platform research. IEEE Access, 12, 155218-155233.

[9] Bao, S., & Wang, J. (2025). Research on the methodology of personalized recommender systems based on multimodal knowledge graphs. Natural Language Processing Journal, 13, 100193.

[10] Zhai, X., Wang, Y., Liang, L., Wang, K., Pei, F., & Fu, E. Y. (2025). Personalized

e-learning resource recommendation using multimodal-enhanced collaborative filtering. Knowledge-Based Systems, 319, 113605.

[11] Shafiq, D. A., Jhanjhi, N. Z., & Abdullah, A. (2022). Load balancing techniques in cloud computing environment: A review. Journal of King Saud University-Computer and Information Sciences, 34(7), 3910-3933.

[12] Chhabra, S., & Singh, A. K. (2021). Dynamic resource allocation method for load balance scheduling over cloud data center networks. Journal of Web Engineering, 20(8), 2269-2284.

[13] Cao, H. (2023). The analysis of edge computing combined with cloud computing in strategy optimization of music educational resource scheduling. International Journal of System Assurance Engineering and Management, 14(1), 165-175.

[14] Su, M., Wang, G., & Choo, K. K. R. (2022). Prediction-based resource deployment and task scheduling in edge-cloud collaborative computing. Wireless Communications and Mobile Computing, 2022, 2568503.

[15] Lv, Z. (2022). Design of cross-source education information classification model based on cloud computing technology. Advances in Multimedia, 2022, 7649317.

[16] Huang, H., & Yang, X. (2024). Cloud computing-based method for optimal allocation of college network course education resources. International Journal of Continuing Engineering Education and Life Long Learning, 34(5), 501-515.

[17] Zhang, Y. (2022). Analyzing the construction of university ELT resource base using cloud platform. Mobile Information Systems, 2022, 4986923.

[18] Jin, N. (2022). Students' ubiquitous learning model and resource sharing of English education based on cloud computing. Wireless Communications and Mobile Computing, 2022, 4451210.

[19] He, Y. (2022). English multimedia online teaching resource processing system based on intelligent cloud platform. Mobile Information Systems, 2022, 9952782.

[20] Wang, J., & Li, W. (2021). The construction of a digital resource library of English for higher education based on a cloud platform. Scientific Programming, 2021, 4591780.

[21] Penney, D., Li, B., Chen, L., & Sydir, J. J. (2023). RAPID: Enabling fast online policy learning in dynamic public cloud environments. Neurocomputing, 558, 126737.

[22] Zhang, J., Ning, Z., Waqas, M., Alasmary, H., Tu, S., & Chen, S. (2023). Hybrid edge-cloud collaborator resource scheduling approach based on deep reinforcement learning and multiobjective optimization. IEEE Transactions on Computers, 73(1), 192-205.

[23] Bodra, D., & Khairnar, S. (2025). Machine learning-based cloud resource allocation algorithms: A comprehensive comparative review. Frontiers in Computer Science, 7, 1678976.

[24] Pan, J., Wei, Y., Meng, L., & Meng, X. (2025). A dual scheduling framework for task and resource allocation in clouds using deep reinforcement learning. Journal of King Saud University Computer and Information Sciences, 37, 81.

[25] Mohiuddin, K., Fatima, H., Khan, M. A., Khaleel, M. A., Begum, Z., Khan, S. A., et al. (2023). Design of a novel edge-centric cloud architecture for m-learning performance effectiveness by leveraging distributed computing paradigms' potentials. SAGE Open, 13(3), 21582440231190337.

[26] Miao, Y. (2022). University educational administration management platform integrating distributed real-time cloud computing system. Mathematical Problems in Engineering, 2022, 1378931.

[27] Devi, N., Dalal, S., Solanki, K., Dalal, S., Lilhore, U. K., Simaiya, S., et al. (2024). A systematic literature review for load balancing and task scheduling techniques in cloud computing. Artificial Intelligence Review, 57(10), 276.

[28] Li, F., & Wang, C. (2023). Artificial intelligence and edge computing for teaching quality evaluation based on 5G-enabled wireless communication technology. Journal of Cloud Computing, 12, 45.

[29] Xie, H., Li, C., Ye, Z., Zhao, T., Xu, H., & Du, J. (2025). Cloud resource scheduling using multi-strategy fused honey badger algorithm. Big Data, 13(1), 59-72.

[30] Mostafa, R. R., Chhabra, A., Khedr, A. M., & Hashim, F. A. (2024). Boosting white shark optimizer for global optimization and cloud scheduling problem. Neural Computing and Applications, 36(18), 10853-10879.