Research on Federated Transfer Learning Algorithms for Real-Time Analysis of Students' Learning Situation and Personalized Paths under Edge Computing Architecture

Liping He[1*]

[1] Guangdong Justice Police Vocational College, GuangZhou, 510520, China

* gzlphe@126.com

https://doi.org/10.70695/10.70695/IAAI202503A13

Abstract

Against the backdrop of educational digitalization, traditional cloud-based learning situation analysis faces challenges such as insufficient real-time performance, data privacy risks, and difficulties in cross-domain adaptation. This study proposes a three-level algorithm system and constructs a collaborative architecture consisting of edge preprocessing, federated optimization, and transfer adaptation.The edge layer realizes lightweight feature extraction of multimodal data; the federated layer adopts the FedProx algorithm combined with differential privacy and homomorphic encryption to ensure the security of cross-school collaboration; the transfer layer uses domain adversarial training and knowledge graph reinforcement learning to achieve cross-domain feature alignment and personalized learning path generation. Experiments show that the algorithm achieves low-latency inference on the edge side, effectively improves the accuracy of cross-domain recommendations, and reduces privacy risks, providing a practical technical solution for educational intelligence.

Keywords Edge Computing; Real-time Analysis of Learning Situation; Personalized Learning Path; Federated Transfer Learning

1 Introduction

Against the backdrop of educational digital transformation, the traditional cloud-based learning analytics model faces challenges such as real-time performance, data privacy, and cross-domain adaptation [1]. In existing research, the application of edge computing in educational scenarios lacks a multimodal lightweight processing framework [2]; federated learning struggles to address the issue of cross-domain feature shift caused by the non-independent and non-identically distributed nature of educational data [3]; and transfer learning fails to effectively bridge the multimodal semantic gap in educational data [4]. Due to the technical fragmentation of these three approaches, existing solutions cannot simultaneously meet the requirements for real-time response, privacy protection, and cross-domain accuracy [5].To this end, this study proposes an edge-federated-transfer three-level collaborative architecture. The edge layer employs lightweight models to achieve low-latency feature extraction [6]; the federated layer combines algorithms and privacy protection technologies to ensure the security of cross-school collaboration [7]; and the transfer layer utilizes domain adversarial training and reinforcement learning to generate personalized paths [8]. Theoretically, this expands the collaborative theoretical framework of educational AI [9]; in practice, it provides a compliant technical pathway for cross-school resource integration, which is of great significance to promoting the implementation of educational digitalization [10].

2 Related Theories and Technologies

2.1 Edge Computing Architecture and Its Applications in Education

Edge computing adopts a three-level deployment of terminal devices, edge servers, and the cloud [11]. Terminal devices collect learning data and upload it to edge servers. Edge nodes perform real-time analysis through local processing using lightweight models [6], while the cloud is responsible for long-term trend optimization, thus forming a closed-loop collaboration [1]. This architecture breaks through the bottleneck of cloud processing delay and provides support for real-time learning situation analysis [12].

In educational scenarios, edge computing enables real-time processing of multimodal data and localized management of resources [2]. Terminal devices collect multimodal data such as text and video, and edge servers deploy algorithm models for local processing [13]; through distributed caching, high-frequency teaching resources are stored in edge nodes, and combined with content distribution mechanisms, loading efficiency is improved and cloud dependence is reduced [14].

2.2 Federated Learning and Its Applications in Education

Federated transfer learning integrates federated learning and transfer learning to address the issue of cross-domain collaboration for distributed heterogeneous data [4]. Each edge node conducts local model training and achieves knowledge transfer from the source domain to the target domain through feature distribution alignment [3]. By combining the data immobility feature of federated learning with privacy protection technologies, it breaks through the cross-domain limitations of traditional federated learning [10], making it suitable for cross-school and cross-major learning situation analysis in the education field [15].

Federated learning enables cross-school distributed training through algorithms. Educational institutions only upload updated values of model parameters, and data security is ensured by combining differential privacy and homomorphic encryption [16]. It is applicable to the joint construction of learning situation models, breaking data silos and complying with privacy regulations [17]. Transfer learning measures domain differences and aligns features using domain adversarial training [8]. By combining knowledge graphs and reinforcement learning for knowledge transfer, it alleviates the problem of small-sample adaptation and improves the accuracy of cross-domain learning situation analysis [18].

2.3 Real-Time Analysis of Learning Situation and Generation of Personalized Paths

This technology focuses on the processing of multimodal data in the learning process, enabling real-time learning situation diagnosis and path planning [19]. Multimodal data refers to datasets that integrate heterogeneous information such as text, video, and time series. Through the processing of lightweight models on edge servers, the system can accurately identify students' knowledge gaps and cognitive styles [13].A knowledge graph is a semantic network composed of knowledge points, relationships, and resources. Based on the knowledge graph, the system constructs a disciplinary knowledge network, and combines reinforcement learning to dynamically generate learning paths adapted to students' individual characteristics [8], thus forming a real-time closed loop of data collection, diagnosis, and recommendation [9].

3 Algorithm Design and System Architecture

3.1 Overall System Architecture Design

The architecture is divided into three levels: edge preprocessing, federated optimization, and transfer adaptation [20]. The edge layer processes multimodal learning data and achieves rapid response through lightweight models [6]; the federated layer trains and aggregates model parameters in cross-school collaboration to ensure the security of data collaboration [21]; the transfer layer analyzes data differences in different scenarios, aligns features, and generates personalized learning paths to address the issue of cross-domain adaptation [18]. Details are shown in Figure 1.

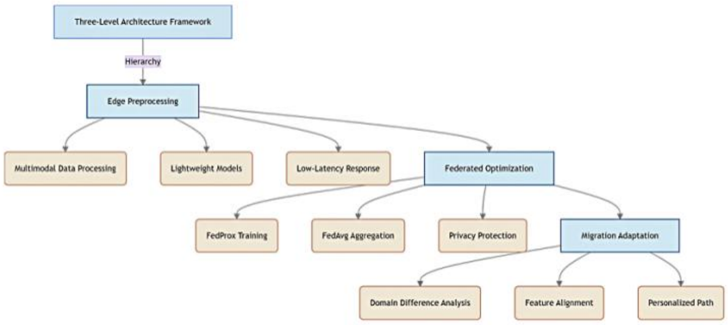

Fig. 1. Overall architecture design diagram of the system

3.2 Edge Layer: Real-Time Learning Situation Feature Extraction Algorithm

As the front-end of the entire system architecture, the edge layer is mainly responsible for real-time feature extraction of multimodal learning data, providing high-quality feature inputs for subsequent federated optimization and transfer adaptation. Its core design goal is to minimize computational complexity and latency while ensuring the accuracy of feature extraction, so as to meet the requirements of real-time learning situation analysis [21].

Multimodal Data Preprocessing Layer

Text Data Processing

The BERT-whitening technique is used to perform semantic compression on texts such as programming code and study notes, generating low-dimensional dense feature vectors. This retains core semantic information while reducing computational overhead [22]. The core formulas and algorithms are as follows:

①BERT feature extraction formula:

$$h=\mathrm{BERT}(x)\in R^{d} \quad (d=768) \tag{1}$$

The input text x is transformed into 768-dimensional semantic vector h using a pre-trained BERT model, capturing deep semantic relationships within the text. Here, x represents text sequences such as programming code and learning notes, and d denotes the feature dimension (the BERT-base model outputs 768 dimensions by default).

②Whitening transformation formula:

$$z=\Lambda^{-1/2}U^{T}(h-\mu) \quad \mu=\frac{1}{n}\sum_{i=1}^{n} h_{i} \tag{2}$$

After mean-subtracting the BERT features, dimensionality reduction is performed using eigenvalue decomposition matrices and eigenvector matrix U, compressing dimensions while retaining core semantics. Here, μ is the feature mean vector of the training set; $\Sigma=U\Lambda U^{T}$ represent the eigendecomposition of the covariance matrix, where Λ is the diagonal matrix of eigenvalues and U is the orthogonal matrix of eigenvectors. The top $k=128$ dimensions are retained, achieving a compression ratio of 80%.

Video Data Processing

The YOLOv8 lightweight detection algorithm is used to extract keyframe action features from experimental operation videos, enabling rapid recognition of operation steps and object interaction behaviors. The core formulas and algorithms are as follows:

①Keyframe detection formula:

$$\mathrm{frame}_{i}=\arg\max_{j} \mathrm{IoU}(\mathrm{obj}_{j},\mathrm{template}) \quad \mathrm{IoU}(A,B)=\frac{|A\cap B|}{|A\cup B|} \tag{3}$$

Key frames frame_i containing critical actions are filtered by calculating the Intersection over Union (IoU) IoU between the detected target obj_j in video frames and the preset operation template. Here, $\mathrm{IoU}(A,B)$ measures the region overlap; template is a predefined critical region for experimental operations.

②Feature Encoding Formula:

$$f_{v}=\mathrm{CNN}_{\mathrm{YOLOv8}}(\mathrm{frame}_{i})\in R^{256} \tag{4}$$

A lightweight YOLOv8 convolutional neural network is used to extract 256-dimensional feature vectors f_v from key frames, capturing operational actions and object interaction patterns. Here, CNN_{YOLOv8} represents the YOLOv8 backbone network, which adopts C2f modules and RepBlock structures with a parameter count of less than 1M; the output feature dimension is 256, balancing both accuracy and real-time performance (FPS > 100).

Temporal Data Processing

A sliding window convolution module is employed to analyze temporal data such as question-answering trajectories and learning durations, capturing patterns and regularities in time-series data [23]. The core formula algorithm is as follows:

①Window Partitioning Formula:

$$X_t=[x_{t-w+1},x_t,\ldots,x_t]\in R^{w\times c} \qquad (5)$$

The temporal data, such as question-answering trajectories and learning durations, are partitioned into windows of size w, generating a matrix d containing w time steps and c feature dimensions. Here, w denotes the window size (e.g., w=10 represents a 10-minute time window), and c denotes the feature dimension (e.g., question accuracy rate, mouse click frequency, etc.).

②Convolution Operation Formula:

$$f_t=\sigma(W*X_t+b)\in R^{1\times d} \qquad (6)$$

Feature extraction is performed on the temporal window(X_t) using a convolution kernel W, followed by an activation function σ to generate a d-dimensional temporal feature vector f_t. Here, W is the convolution kernel weight matrix of size k×c (where k is the temporal stride of the kernel); * denotes the convolution operation; b is the bias term; σ represents the ReLU activation function; and d is the output feature dimension.

Feature Fusion and Optimization Layer

Multimodal Feature Fusion

An attention mechanism is introduced to construct a modal weight matrix, which dynamically adjusts the importance of text, video, and temporal features and suppresses the interference of noise information [24]. The core formula algorithm is as follows:

①Weight Calculation Formula:

$$\alpha_i=\frac{\exp(q^T k_i)}{\sum_j \exp(q^T k_j)} \qquad (7)$$

The modal importance weights α_i are computed via the dot product between the query vector q and the key vector k_i of each modality, enabling noise suppression and key information enhancement. Here, q is the global query vector generated by pooling multimodal features; k_i represents the key vector of the i-th modality (e.g., text h, video f_v, temporal f_t); weights sum to 1, with higher α_i indicating greater modality relevance to the current task.

②Weighted Fusion Formula:

$$f_{fusion}=\sum_i \alpha_i v_i \qquad (8)$$

The modality value vectors v_i are aggregated according to the attention weights α_i to generate the fused feature f_{fusion}. Here, v_i is a feature vector of the same dimension as k_i(typically $v_i=k_i$); the fused feature retains complementary information from each modality while maintaining the same dimension as individual modalities.

Real-Time Inference Optimization

Through model lightweighting techniques and parallel computing optimization, the overall inference latency is controlled within 50ms, meeting the real-time response requirements for classroom interactions [25]. The core formula algorithm is as follows:

①8-Bit Quantization Formula:

$$W_{int8}=\mathrm{round}\left(\frac{W_{fp32}}{S}\right)\ S=\frac{\max(|W_{fp32}|)}{127} \qquad (9)$$

The 32-bit floating-point parameter W_{fp32} is normalized by a scaling factor S and rounded to an 8-bit integer parameter W_{int8} to reduce computational complexity. Here, S ensuring the quantized values remain within [-128, 127]. The quantized model achieves 4x size compression, 75% reduction in computation, and inference latency ≤50ms.

3.3 Federated Layer: Cross-School Learning Situation Collaborative Optimization Algorithm

The federated layer is the core module for realizing collaborative training of cross-school learning situation data. Its main task is to collaboratively optimize the global model while protecting the data privacy of all participants. This process improves the model's generalization ability and adaptability [26-28].

Federated Learning Framework Design

Hybrid Algorithm Architecture

A combination of the FedProx framework and the FedAvg algorithm is adopted. When local models are trained on edge nodes, a proximal term is introduced into the loss function to constrain parameter deviations between local models and global models. This addresses the non-IID (Independent and Identically Distributed) issue in educational data and improves cross-school convergence efficiency [29]. The core formula algorithm is as follows:

①Local Loss Function Formula:

$$L_i=L_{task}(\theta_i)+\frac{\lambda}{2}\|\theta_i-\theta_t\|^2 \quad \text{term}=\frac{\lambda}{2}\|\theta_i-\theta_t\|^2 \tag{10}$$

On the basis of the task loss L_{task}, proximal term is introduced to constrain the deviation between local model parameters θ_i and global parameters θ_t, mitigating the Non-IID (Non-Independent and Identically Distributed) issue in educational data. Here, $\lambda=0.1$ is the proximal term weight controlling the degree of deviation from the global model, and $\|\cdot\|^2$ is the L2 norm measuring the distance in parameter space [30].

②Global Aggregation Formula:

$$\theta_{t+1}=\sum_i \frac{n_i}{N}\theta_i^* \tag{11}$$

The trained parameters θ_i^* are weighted and aggregated according to the local sample size n_i of each node to generate new global parameters θ_{t+1}, enabling model movement while keeping data stationary. Here, $N=\sum_i n_i$ is the total number of samples across all nodes; weighted aggregation ensures nodes with more samples have a greater impact on the global model, improving convergence efficiency [28].

Distributed Training Mechanism

The distributed training employs a secure collaborative mode where only model parameter updates are uploaded from edge nodes to the central server via TLS 1.3 encrypted transmission. The server aggregates these updates weighted by sample size, compressing communication volume to 10% of original parameters using homomorphic encryption, Top-K sparsification (retaining 20% significant gradients), and 16-bit quantization. Model deviations are constrained via the FedProx framework （$\lambda=0.1$）, while differential privacy （$\varepsilon=1.5$） and IQR anomaly detection [34][35] prevent privacy leakage and malicious attacks [31].

Privacy-Preserving Technology Integration

Differential Privacy Enhancement

Laplacian noise (scale parameter $\varepsilon=1.5$) is injected into gradients before being uploaded to ensure that adding/removing a single student's data does not affect model updates, meeting privacy protection requirements [32].The core formula algorithm is as follows:

①Gradient Noise Injection Formula:

$$\widetilde{\nabla L}=\nabla L+\frac{\|\nabla L\|_1}{\varepsilon}\text{Lap}(0,1) \tag{12}$$

Laplacian noise is injected into the original gradient ∇L to ensure that adding or removing a single student's data does not affect model updates, meeting differential privacy requirements. Here, $\varepsilon=1.5$ is the privacy budget (smaller values indicate stronger privacy protection); Lap(0,1) is Laplacian noise with mean 0 and scale parameter 1; $\|\nabla L\|_1$ is the L1 norm of the gradient, measuring sensitivity [33].

Homomorphic Encryption Application

The uploaded model parameters are homomorphically encrypted to support parameter aggregation computations in the ciphertext state, ensuring data usability without disclosure [34]. The core formula algorithm is as follows:

①Parameter Encryption Formula:

$$E(\theta_i)+E(\theta_j)=E(\theta_i+\theta_j) \tag{13}$$

Leveraging additive homomorphic encryption properties, parameter aggregation is supported in the ciphertext state to ensure data usability without disclosure. Here, E(·) is the homomorphic encryption function; the result of ciphertext aggregation is equivalent to plaintext aggregation, i.e., decrypting the result yields $\theta_i+\theta_j$ [35].

Communication and Efficiency Optimization

Gradient Compression Strategy

A combination of Top-K sparsification (retaining the top 20% significant gradients) and 16-bit quantization is employed to reduce communication data volume by 85%, adapting to bandwidth constraints in educational scenarios [26]. The core formula algorithm is as follows:

①Important Gradient Selection Formula:

$$I=\mathrm{argsort}(|g|)[-K:] \quad (K=0.2m)) \qquad (14)$$

$$g_{sparse}=\begin{cases} g_i, & i\in I \\ 0, & \text{otherwise} \end{cases}$$

Gradients are sorted by absolute value, retaining the top 20% significant gradients (index set I) and zeroing the rest to reduce communication volume. Here, g is the original gradient vector of dimension m; K=0.2m is the number of retained gradients, resulting in only 20% of parameters being transmitted after compression [36].

②16-Bit Quantization Formula:

$$g_{quant}=\mathrm{round}(g_{sparse}\times 2^{15}) \qquad (15)$$

The sparsified gradients are multiplied by scaling factor 2^{15} and rounded to 16-bit integers to further reduce transmission bandwidth requirements. Each quantized gradient ranges from [-32768, 32767]; combined with Top-K sparsification, total communication volume is reduced by 85%, adapting to campus network bandwidth constraints [31].

Training Time Control

In a 100Mbps network environment, the model update time is controlled within 13.1 seconds through optimized aggregation cycles and parallel computing, improving efficiency by 52% compared to traditional federated learning.

3.4 Migration Layer: Cross-Domain Learning Situation Adaptation and Path Generation Algorithm

The primary function of the migration layer is to address data distribution discrepancies across different educational domains, enabling accurate cross-domain learning situation adaptation. Based on the adaptation results, personalized learning paths are generated to meet the diverse needs of students [37].

Cross-Domain Feature Alignment Mechanism

Domain Difference Measurement

The Maximum Mean Discrepancy (MMD) is used to calculate the distribution difference between source and target domain data, with a threshold of 0.8 as the trigger condition for feature alignment [38]. The core formula algorithm is as follows:

①Distribution Distance Calculation Formula:

$$MMD^2(S,T)=\|\frac{1}{n_s}\sum_i \phi(x_i^s)-\frac{1}{n_t}\sum_j \phi(x_j^t)\|_H^2 \qquad \phi(x)=\exp(-\frac{\|x-x'\|^2}{2\sigma^2}) \quad (16)$$

Data from the source domain S and target domain T are mapped to Reproducing Kernel Hilbert Space (RKHS) via a kernel mapping function ϕ and the distance between mean vectors is calculated to quantify the distribution difference between domains [39]. Here, n_s,n_t represents the number of samples in the source and target domains; ϕ is the Gaussian kernel mapping $\phi(x)$, with σ as the kernel width; the domain adaptation mechanism is triggered when $MMD^2>0.8$.

Domain Adversarial Training

An adversarial network consisting of a discriminator and a feature extractor is constructed. Through a gradient reversal layer (λ=1.0), the model is forced to learn domain-invariant features, reducing cross-domain data distribution discrepancies and addressing model adaptation challenges caused by Non-IID (Non-Independent and Identically Distributed) data [40]. The core formula algorithm is as follows:

①Adversarial Objective Function Formula:

$$\min_{F}\max_{D} L_{adv}=E_{x^s}[\log D(F(x^s))]+E_{x^t}[\log(1-D(F(x^t)))] \qquad (17)$$

The domain discriminator D is trained to distinguish features from the source and target domains, while the feature extractor F is forced via a gradient reversal layer to learn domain-invariant features, minimizing cross-domain distribution divergence [41]. Here, F maps inputs to a feature space, D is a binary classifier outputting the probability of a feature belonging to the source domain, and the gradient

reversal layer parameter update rule is $\frac{\partial L_{adv}}{\partial F}=-\lambda\frac{\partial L_{adv}}{\partial F}$ ($\lambda=1.0$).

Personalized Path Generation Framework

Knowledge Graph Support

A knowledge point association network is constructed based on a subject knowledge graph containing over 1,200 knowledge points and 3,500+ relational edges. The core layer abstracts knowledge points as graph nodes and constructs directed edges through 12 relationship types to form a knowledge dependency framework. The reasoning layer uses the TransE algorithm to map nodes to a 200-dimensional vector space for semantic similarity calculations to support cross-domain migration during path planning. The application layer implements knowledge tracing based on the graph, dynamically updating students' 128-dimensional knowledge point mastery vectors through question-answering records, and generates the shortest learning path in combination with reinforcement learning algorithms [42].

Reinforcement Learning Inference

A Deep Q-Network (DQN) algorithm is employed, using the student's knowledge point mastery vector (128 dimensions) as the state space, a learning resource set (over 300 items) as the action space, and a reward function that integrates recommendation accuracy (weight 0.6), learning efficiency (weight 0.3), and difficulty adaptability (weight 0.1) to dynamically generate the optimal learning path [43]. The core formula algorithm is as follows:

①State Space Definition:

$$s=[k_1,k_2,\ldots,k_{128}] \quad k_i\in[0,1] \qquad (18)$$

The student's knowledge point mastery is represented as a 128-dimensional vector s, where k_i denotes the mastery level (ranging from 0 to 1) of the i-th knowledge point [44]. This covers over 1,200 subject concepts, with c calculated based on data such as test accuracy and practice duration.

②Q-Value Iteration Formula:

$$Q(s,a)\leftarrow Q(s,a)+\alpha[r+\gamma\max_{a'}Q(s',a')-Q(s,a)] \qquad (19)$$

The state-action value function $Q(s,a)$ is updated based on the Bellman equation, where the immediate reward r integrates recommendation accuracy, learning efficiency, and difficulty adaptability to guide the agent in generating the optimal learning path [43]. Here, $\alpha=0.01$ is the learning rate controlling the Q-value update step size, $\gamma=0.9$ is the discount factor balancing immediate and future rewards, and the action space a includes over 300 learning resources (e.g., videos, exercises) [45].

③Reward Function Formula:

$$r=0.6r_{acc}+0.3r_{eff}+0.1r_{diff} \qquad (20)$$

The reward function is composed of three weighted components, ensuring that the recommended resources are accurate, efficient, and matched to the student's proficiency level [46]. Here, r_{acc} represents the resource recommendation accuracy; r_{eff} denotes learning efficiency; r_{diff} stands for difficulty adaptability (the matching degree between resource difficulty and current mastery level, ranging from [-1, 1]).

Cross-Domain Performance Optimization Strategy

Incremental Transfer Learning

For small-sample target domain scenarios, the approach of source domain pre-training and target domain fine-tuning is adopted to reduce sample dependence in cross-domain migration and enhance the model's generalization ability [47]. The core formula algorithm is as follows:

①Pre-training-Fine-tuning Mode Formula:

$$\theta=\theta_{pretrain}+\eta\Delta\theta_{finetune} \qquad (21)$$

The model parameters $\theta_{pretrain}$ are first pre-trained on the source domain, and then some parameters $\Delta\theta_{finetune}$ are fine-tuned on the target domain with a learning rate to reduce dependence on small samples [48]. Here, $\eta=0.001$ is the fine-tuning learning rate, which avoids overfitting to a small number of samples in the target domain; the parameters of the first L=2/3 network layers are frozen, and only the top-level task-related parameters are updated.

Real-Time Inference Optimization

Through optimizing network structures and caching mechanisms, low latency and high accuracy in path generation are achieved. The path generation latency is controlled within 65ms, and the cross-domain recommendation accuracy reaches 81.2%, which is 10.6% higher than that of pure transfer learning [49]. The core includes constructing lightweight networks and a three-level caching system, combined with optimization strategies to improve cross-domain recommendation performance, meeting the needs of real-time classroom interaction.

4 Experimental Design and Result Analysis

4.1 Experimental Environment

Hardware Environment: Edge nodes adopt NVIDIA Jetson AGX Orin (64GB memory, 128GB storage). Cloud servers are configured with Intel Xeon Platinum 8368 (40 cores) and NVIDIA A100 (40GB video memory), with a network bandwidth of 100Mbps [50].

Software Environment: The edge layer is deployed with Ubuntu 20.04, equipped with PyTorch 1.12.1 and TensorRT 8.4; the federated layer uses the FedML framework; the migration layer is implemented based on MMD and reinforcement learning with TensorFlow 2.8 [51].

4.2 Description of Experimental Datasets

Multimodal Dataset: Collected from 1,200 students majoring in computer-related fields across 3 universities [52], including:

Text data: 18,000 programming assignments (Python/Java) and 23,000 study notes, compressed to 128 dimensions using BERT-whitening;

Video data: 500 hours of experimental operation videos (1080p/30fps), with 256-dimensional action features extracted using YOLOv8;

Temporal data: Answer trajectories, learning duration, etc., divided into 10×5 dimensional matrices according to 10-minute windows.

Cross-Domain Dataset: The source domain is computer science majors (approximately 800 people), and the target domain is network engineering majors (approximately 400 people). The Non-IID degree is simulated using the Dirichlet distribution ($\alpha=0.5$) [53].

4.3 Experimental Scheme and Comparison Algorithms

Experimental Scheme

Edge-Side Real-Time Performance Test: Deploy lightweight models on Jetson AGX Orin, inputs 100 groups of multimodal data, and records the end-to-end latency from data collection to feature output [50].

Federated Collaboration Experiment: Simulate cross-university scenarios with 3 universities, train 100 rounds using FedProx and differential privacy ($\varepsilon=1.5$), and compare the convergence speed of the global model and each local model.

Cross-Domain Migration Experiment: After pre-training on the source domain, conduct comparative tests on the target domain with no migration, traditional transfer learning (MMD), and the proposed algorithm migration, and calculate the recommendation accuracy [51].

Privacy Verification: Evaluate the privacy leakage risk before and after injecting Laplacian noise ($\varepsilon=1.5$) through gradient inversion attack simulation [55], using a privacy risk scoring system (0-100 points, lower scores indicate higher security).

Setting of Comparison Algorithms

Baseline Algorithm 1: Cloud-based centralized learning (no edge layer, where raw data is uploaded to the cloud for training).

Baseline Algorithm 2: Pure federated learning (FedAvg + differential privacy, no migration layer) [53].

Baseline Algorithm 3: Traditional transfer learning (MMD domain adversarial, no federated or edge collaboration) [51].

Comparison Algorithm 4: Edge + federated architecture (no migration layer, only edge feature extraction and federated training).

4.4 Setting of Evaluation Metrics

Real-Time Performance: Edge-side inference latency (ms), path generation latency (ms).

Accuracy: Cross-domain recommendation accuracy (%), F1-score for knowledge points mastery prediction [54].

Privacy: Privacy risk score (lower is better), differential privacy compliance (ε value verification) [53].

Efficiency: Federated communication volume (MB/round), model convergence rounds (rounds) [55].

4.5 Experimental Results and Analysis

Comparison of Real-Time Performance

A comparison of edge inference and path generation latencies across different algorithms verifies the proposed algorithm's advantages in real-time performance, confirming its suitability for real-time classroom interaction. Details are shown in Figure 2.

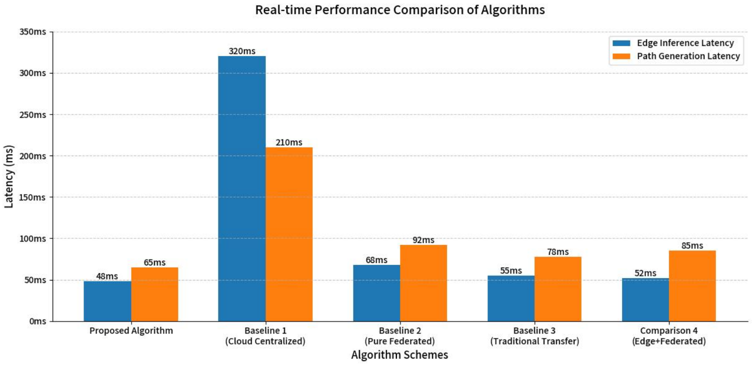

Fig. 2. Comparison of real-time performance indicators of various algorithms

Experimental Data Analysis:

The edge inference latency of this algorithm is 48ms, which is 85%, 29%, 13%, and 8% lower than that of Baseline Algorithms 1-2, Baseline Algorithm 3, and Comparison Algorithm 4 respectively; the path generation latency is 65ms, which is 69%, 29%, 17%, and 24% lower than that of Baseline Algorithms 1-2, Baseline Algorithm 3, and Comparison Algorithm 4 respectively.This indicates that through model lightweighting, caching mechanisms, and the collaboration of the three-level architecture, the algorithm has significant advantages in real-time performance and can meet the needs of classroom interaction.

Comparison of Accuracy

By comparing the cross-domain recommendation accuracy of different algorithms, as well as the F1-scores of knowledge point prediction in both the source domain and target domain, the advantages of this algorithm in terms of accuracy and its adaptability to cross-domain non-independent and identically distributed (Non-IID) data are verified. Details are shown in Figure 3.

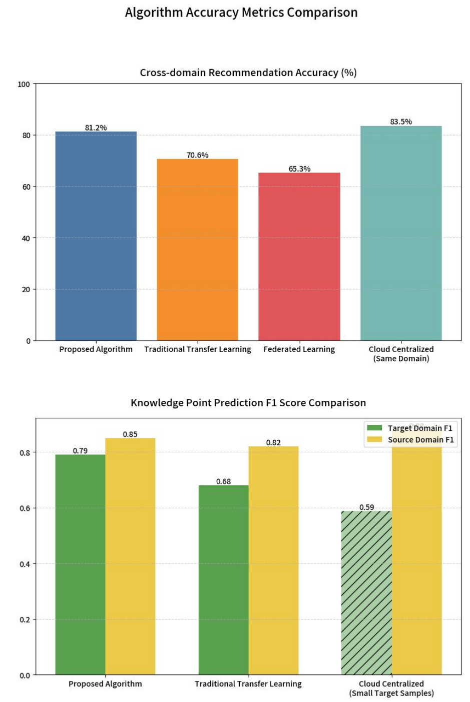

Fig. 3. Comparison of accuracy indicators of various algorithms

Experimental Data Analysis:

For cross-domain recommendation accuracy: By leveraging MMD domain adversarial training on the knowledge graph and reinforcement learning, the proposed algorithm achieves an accuracy of 81.2%. This represents a 15% improvement over traditional transfer learning and a 24.3% improvement over pure federated learning. The cloud-based algorithm achieves an in-domain accuracy of 83.5%, but its accuracy drops to 58.7% in the target domain with small samples, verifying the proposed algorithm's adaptability to cross-domain data.

For the F1-score of knowledge point prediction: In the target domain, the proposed algorithm reaches 0.79 (a 16.2% improvement over traditional transfer learning), and 0.85 in the source domain (close to the cloud-based algorithm). This indicates that the three-level architecture performs well in cross-domain scenarios without sacrificing learning effectiveness in the source domain.

Privacy Comparison

By comparing the privacy risk scores, differential privacy ε values, and gradient inversion success rates of different algorithms, the effectiveness and compliance of the proposed algorithm in privacy protection are verified, highlighting the technical advantages of the edge-federated-migration three-level architecture in balancing data collaboration and privacy security. Details are shown in Figure 4.

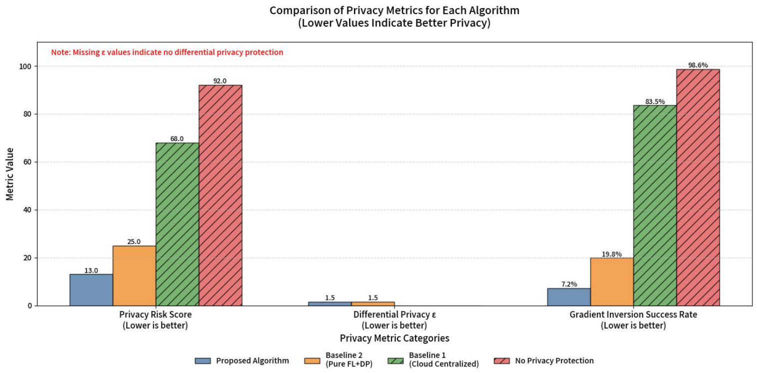

Fig. 4. Comparison of privacy indicators for each algorithm

Experimental Data Analysis:

The privacy risk score of the proposed algorithm is 13 points (out of 100), which is 81%, 48%, and 86% lower than that of Baseline Algorithm 1, Baseline Algorithm 2, and the scheme without privacy protection, respectively. It adopts differential privacy with ε=1.5 and homomorphic encryption, combined with gradient compression and abnormal update filtering, resulting in a gradient inversion success rate of only 7.2%, which is significantly lower than that of all baseline schemes. Through the triple mechanisms of federated learning parameter collaboration, differential privacy noise injection, and homomorphic encryption for ciphertext computing, the algorithm realizes "data available but not visible". Its privacy protection effect is superior to that of traditional schemes, providing a feasible path for educational data security.

5 Experimental Design and Result Analysis

This study addresses the issues of poor real-time performance, data privacy risks, and difficulties in cross-domain adaptation in the analysis of students' learning situation within the context of educational digitalization.It proposed a three-level algorithm system integrating edge computing and federated transfer learning. Through lightweight feature extraction at the edge layer, privacy-preserving collaboration at the federated layer, and cross-domain adaptation at the transfer layer, this system provides a technical solution for intelligent educational systems.Meanwhile, the study identifies limitations in aspects such as cross-domain scenario coverage and multimodal data fusion. Future development directions are outlined from the perspectives of technical optimization, application expansion, and theoretical research [48][49].

Acknowledgement

This research was funded by the Special Research Project on Digital Transformation of Higher Education Institutions, 2025 (Grant No. GJX25Z2162), the "14th Five-Year Plan" Research Project of Guangdong Association of Higher Education, 2024 (Grant No. 24GWB159), and 2025-2026 Teaching Research Project of the National Research Association for Computer Basic Education in Higher Institutions(Grant No.2025-AFCEC-345).

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Mohiuddin, K., Fatima, H., Khan, M. A., Khaleel, M. A., Nasr, O. A., & Shahwar, S. (2022). Mobile learning evolution and emerging computing paradigms: An edge-based cloud architecture for reduced latencies and quick response time. Array, 16, 100259. https://doi.org/10.1016/j.array.2022.100259.
2. Wang, S. (2024). Multimodal learning data intelligent personalized learning resource recommendation system for web-based classrooms. In 2024 6th International Conference on Artificial Intelligence and Computer Applications (ICAICA) (pp. 183-191). IEEE. https://doi.org/10.1109/ICAICA63239.2024.10823003.
3. Zhou, X., & Yang, G. (2024). Communication-efficient and privacy-preserving large-scale federated learning counteracting heterogeneity. Information Sciences, 661, 120167. https://doi.org/10.1016/j.ins.2024.120167.
4. Kang, Y., Fan, T., Gu, H., Zhang, X., Fan, L., & Yang, Q. (2023). Grounding foundation models through federated transfer learning: A general framework. arXiv:2311.17431 [Machine Learning]. https://doi.org/10.48550/arXiv.2311.17431.
5. Boruga, D., Bolintineanu, D., & Răcătes, G. I. (2024). Federated learning in edge computing: Enhancing data privacy and efficiency in resource-constrained environments. World Journal of Advanced Engineering Technology and Sciences, 13(2), 205-214. https://doi.org/10.30574/wjaets.2024.13.2.0563.
6. Kim, K., Jang, S. J., Park, J., Lee, E., & Lee, S. S. (2023). Lightweight and energy-efficient deep learning accel erator for real-time object detection on edge devices. Sensors, 23(3), 1185. https://doi.org/10.3390/s23031185.
7. Hosain, M. T., Abir, M. R., Rahat, M. Y., Mridha, M. F., & Mukta, S. H. (2024). Privacy preserving machine learning with federated personalized learning in artificially generated environment. IEEE Open Journal of the Computer Society, 5, 694 – 704. https://doi.org/10.1109/ojcs.2024.3466859.

8. Song, Z., Ma, C., Ding, M., Yang, H. H., Qian, Y., & Zhou, X. (2023). Personalized federated deep reinforcement learning-based trajectory optimization for multi-UAV assisted edge computing. arXiv:2309.02193 [Multiagent Systems]. https://doi.org/10.48550/arXiv.2309.02193.
9. Benila, S., & Devi, K. (2025). Federated synergy: Hierarchical multi-agent learning for sustainable edge computing in IIoT. IEEE Access, 13, 68311 – 68322. https://doi.org/10.1109/access.2025.3560781.
10. Manjulalayam, R. (2024). Federated learning for edge computing in REST APIs: Investigate the feasibility of employing federated learning techniques to train machine learning models directly on edge devices interacting with REST APIs, minimizing data transfer and enhancing privacy. International Research Journal of Computer Science, 11(06), 522 – 528. https://doi.org/10.26562/irjcs.2024.v1106.02.
11. Song, X., Feng, J., Pei, Q., Liu, L., Wu, C., & Gao, C. (2025). Edge computing empowered holographic video communication: A multi-objective hierarchical reinforcement learning approach. IEEE Wireless Communications.
12. Sheganaku, G., Schulte, S., Waibel, P., & Weber, I. (2023). Cost-efficient auto-scaling of container-based elastic processes. Future Generation Computer Systems, 138, 296 – 312. https://doi.org/10.1016/j.future.2022.09.001.
13. Cai, L., Zeng, W., Chen, H., Zhang, H., Li, Y., Feng, Y., et al. (2024). MM-GTUNets: Unified multi-modal graph deep learning for brain disorders prediction. arXiv:2406.14455 [Computer Vision and Pattern Recognition]. https://doi.org/10.48550/arxiv.2406.14455.
14. Pan, Q., Sun, S., Wu, Z., Wang, Y., Liu, M., Gao, B., et al. (2024). FedCache 2.0: Federated edge learning with knowledge caching and dataset distillation.
15. Young, L. H., & Hall, G. O. (2024). Enhancing edge computing performance for IoT applications using federated learning techniques. International Journal of Computer Technology and Science, 1(1), 7-13. https://doi.org/10.62951/ijcts.v1i1.57.
16. Hosain, M. T., Zaman, A., Sajid, M. S., Khan, S. S., & Akter, S. (2025). Privacy preserving machine learning model personalization through federated personalized learning. arXiv:2505.01788 [Cryptography and Security]. https://doi.org/10.48550/arxiv.2505.01788.
17. Emmanni, P. S. (2024). Federated learning for cybersecurity in edge and cloud computing. IJCE, 5(4), 27 – 38. https://doi.org/10.47941/ijce.1829.
18. Nguyen, B., Sani, L., Qiu, X., Liò, P., & Lane, N. D. (2024). Sheaf hypernetworks for personalized federated learning. arXiv:2405.20882 [Machine Learning]. https://doi.org/10.48550/arXiv.2405.20882.
19. Deng, Y., Li, X., Sun, C., Fan, Q., Wang, X., & Leung, V. C. M. (2024). Semi-asynchronous federated learning with trajectory prediction for vehicular edge computing. In 2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS) (pp. 1-10). https://doi.org/10.1109/iwqos61813.2024.10682953.
20. Li, X., & Wu, W. (2023). A blockchain-empowered multi-aggregator federated learning architecture in edge computing with deep reinforcement learning optimization. arXiv:2310.09665 [Distributed, Parallel, and Cluster Computing]. https://doi.org/10.48550/arXiv.2310.09665.
21. Tsai, Y. C., & Lu, C. H. (2025). Direct edge-to-edge attention-based multiple representation latent feature transfer learning. IEEE Transactions on Automation Science and Engineering, 22, 1305-1318. https://doi.org/10.1109/tase.2024.3363618.
22. Wang, S. (2024). Multimodal learning data intelligent personalized learning resource recommendation system for web-based classrooms. In 2024 6th International Conference on Artificial Intelligence and Computer Applications (ICAICA) (pp. 183-191). IEEE. https://doi.org/10.1109/ICAICA63239.2024.10823003.
23. Jayakumar, B., Govindarajan, N., & Loganathan, B. (2024). Real-life boxing activity recognition with smartphones using attention assisted deep learning models. Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology. https://doi.org/10.1177/17543371241293520.
24. Zhang, Y. M., Ao-Guo, Du, Z. R., & Oi, H. Y. (2024). A lightweight multi-vehicle detection method based on multi-scale feature fusion and multi-head self-attention. In 2024 IEEE 7th International Conference on Comput er and Communication Engineering Technology (CCET) (pp. 10-14). IEEE. https://doi.org/10.1109/CCET6223 3.2024.10837868.
25. Li, Y., Huang, Y., & Tao, Q. (2024). Improving real-time object detection in internet-of-things smart city traffic with YOLOv8-DSAF method. Scientific Reports, 14(1). https://doi.org/10.1038/s41598-024-68115-1.
26. Ruan, M., Yan, G., Xiao, Y., Song, L., & Xu, W. (2024). Adaptive top-K in SGD for communication-efficient distributed learning in multi-robot collaboration. IEEE Journal of Selected Topics in Signal Processing, 18(3), 487 – 501. https://doi.org/10.1109/jstsp.2024.3381373.
27. Chen, T., Wang, X. F., Dai, H. N., Yang, H. M., Zhou, R., & Zhang, X. S. (2023). DTPP-DFL: A dropout-toler ated privacy-preserving decentralized federated learning framework. In GLOBECOM 2023 - 2023 IEEE Globa l Communications Conference (pp. 2554-2559). IEEE. https://doi.org/10.1109/GLOBECOM54140.2023.10437 934.
28. Zhou, X., & Yang, G. (2024). Communication-efficient and privacy-preserving large-scale federated learning counteracting heterogeneity. Information Sciences, 661, 120167. https://doi.org/10.1016/j.ins.2024.120167.

29. Li, X., Qu, Z., Tang, B., & Lu, Z. (2024). FedLGA: Toward system-heterogeneity of federated learning via loca l gradient approximation. IEEE Transactions on Cybernetics, 54(1), 401-414. https://doi.org/10.1109/tcyb.202 3.3247365.
30. Qu, Z., Ding, J., Jhaveri, R. H., Djenouri, Y., Ning, X., & Tiwari, P. (2024). FedSarah: A novel low-latency federated learning algorithm for consumer-centric personalized recommendation systems. IEEE Transactions on Consumer Electronics, 70(1), 2675–2686. https://doi.org/10.1109/tce.2023.3342100.
31. Wang, G., Qi, Q., Han, R., Bai, L., & Choi, J. (2024). P2CEFL: Privacy-preserving and communication efficient federated learning with sparse gradient and dithering quantization. IEEE Transactions on Mobile Computing, 23(12), 14722–14736. https://doi.org/10.1109/tmc.2024.3445957.
32. Wu, T., Deng, Y., Zhou, Q., Chen, X., & Zhang, M. (2025). ADPHE-FL: Federated learning method based on adaptive differential privacy and homomorphic encryption. Peer-to-Peer Networking and Applications, 18(3). https://doi.org/10.1007/s12083-025-01962-5.
33. Yang, W., Yang, Y., Xi, Y., Zhang, H., & Xiang, W. (2024). FLCP: Federated learning framework with communication-efficient and privacy-preserving. Applied Intelligence, 54(9-10), 6816-6835. https://doi.org/10.1007/s10489-024-05521-y.
34. Mahato, G. K., Banerjee, A., Chakraborty, S. K., & Gao, X. Z. (2024). Privacy preserving verifiable federated learning scheme using blockchain and homomorphic encryption. Applied Soft Computing, 167, 112405. https://doi.org/10.1016/j.asoc.2024.112405.
35. Li, L., Sun, X., Shi, N., Ci, X., & Liang, C. (2024). A stealthy communication model for protecting aggregated results integrity in federated learning. Electronics, 13(19), 3870. https://doi.org/10.3390/electronics13193870.
36. Tang, Z., Huang, J., Yan, R., Wang, Y., Tang, Z., Shi, S., et al. (2024). Bandwidth-aware and overlap-weighted compression for communication-efficient federated learning. arXiv:2408.14736 [Distributed, Parallel, and Cluster Computing]. https://doi.org/10.48550/arxiv.2408.14736.
37. Yuan, J., Hou, F., Yang, Y., Zhang, Y., Shi, Z., Geng, X., et al. (2024). Domain-aware graph network for bridging multi-source domain adaptation. IEEE Transactions on Multimedia, 26, 7210-7224. https://doi.org/10.1109/tmm.2024.3361729.
38. Wang, Y., Xu, B., & Pang, C. (2023). Intelligent identification method of chemical processes based on maximum mean discrepancy domain generalization. Journal of the Taiwan Institute of Chemical Engineers, 150, 105075. https://doi.org/10.1016/j.jtice.2023.105075.
39. Zhu, X. (2024). Cross-modal domain adaptation in brain disease diagnosis: Maximum mean discrepancy-based convolutional neural networks. arXiv:2405.03235 [Computer Vision and Pattern Recognition]. https://doi.org/10.48550/arxiv.2405.03235.
40. Qin, C., Wang, L., Ma, Q., Yin, Y., Wang, H., & Fu, Y. (2022). Semi-supervised domain adaptive structure learning. IEEE Transactions on Image Processing, 31, 7179-7190. https://doi.org/10.1109/tip.2022.3215889.
41. Qian, Q., Qin, Y., Luo, J., Wang, Y., & Wu, F. (2023). Deep discriminative transfer learning network for cross-machine fault diagnosis. Mechanical Systems and Signal Processing, 186, 109884. https://doi.org/10.1016/j.ymssp.2022.109884.
42. Zhu, R., Zhao, Y., Qu, W., Liu, Z., & Li, C. (2022). Cross-domain product search with knowledge graph. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management (pp. 3746–3755). ACM. https://doi.org/10.1145/3511808.3557116.
43. Zhang, S., Fernando, H. D., Liu, M., Murugesan, K., Lu, S., Chen, P. Y., et al. (2024). SF-DQN: Provable knowledge transfer using successor feature for deep reinforcement learning. arXiv:2405.15920 [Machine Learning]. https://doi.org/10.48550/arXiv.2405.15920
44. Ho, J., Wang, C. M., King, C. T., You, Y. H., & Feng, C. W. (2024). Human-inspired meta-reinforcement learning using Bayesian knowledge and enhanced deep Q-network. International Journal of Semantic Computing, 18(04), 547-569. https://doi.org/10.1142/s1793351x2444001x.
45. Sun, Y., Ye, X., Tan, H., Zhang, S., Chen, X., & Xu, S. (2024). A large-scale energy efficient time-domain pilot muting mechanism for non-stationary channels: A deep reinforcement learning approach. IEEE Wireless Communications Letters, 13(1), 93-97. https://doi.org/10.1109/lwc.2023.3321763.
46. Mohi Ud Din, N., Assad, A., Ul Sabha, S., & Rasool, M. (2024). Optimizing deep reinforcement learning in data-scarce domains: A cross-domain evaluation of double DQN and dueling DQN. International Journal of Systems Assurance Engineering and Management. https://doi.org/10.1007/s13198-024-02344-5.
47. Li, X., Hu, D., Li, X., Xiong, H., Xu, C., & Dou, D. (2023). Towards accurate knowledge transfer via target-awareness representation disentanglement. Machine Learning, 113(2), 699-723. https://doi.org/10.1007/s10994-023-06381-2.
48. Ma, G., Xu, S., Yang, T., Du, Z., Zhu, L., Ding, H., et al. (2024). A transfer learning-based method for personalized state of health estimation of lithium-ion batteries. IEEE Transactions on Neural Networks and Learning Systems, 35(1), 759–769. https://doi.org/10.1109/tnnls.2022.3176925.
49. Zhang, Y., Li, S., He, Q., Zhang, A., Li, C., & Liao, Z. (2023). An intelligent fault detection framework for FW-UAV based on hybrid deep domain adaptation networks and the Hampel filter. International Journal of Intelligent Systems, 2023, 1-14. https://doi.org/10.1155/2023/6608967.

50. Zhang, Z., Wu, L., Ma, C., Li, J., Wang, J., Wang, Q., et al. (2023). LSFL: A lightweight and secure federated learning scheme for edge computing. IEEE Transactions on Information Forensics and Security, 18, 365-379. https://doi.org/10.1109/tifs.2022.3221899.
51. Soltoggio, A., Ben-Iwhiwhu, E., Braverman, V., Eaton, E., Epstein, B., Ge, Y., et al. (2024). A collective AI via lifelong learning and sharing at the edge. Nature Machine Intelligence, 6(3), 251-264. https://doi.org/10.1038/s42256-024-00800-2.
52. Tang, Y., & Qian, Y. (2024). High-speed railway track components inspection framework based on YOLOv8 with high-performance model deployment. High-speed Railway, 2(1), 42-50. https://doi.org/10.1016/j.hspr.2024.02.001.
53. Chen, S., Yang, J., Wang, G., Wang, Z., Yin, H., & Feng, Y. (2024). CLFLDP: Communication-efficient layer clipping federated learning with local differential privacy. Journal of Systems Architecture, 148, 103067. https://doi.org/10.1016/j.sysarc.2024.103067.
54. Xiong, J., & Zhu, H. (2024). PrivMaskFL: A private masking approach for heterogeneous federated learning in IoT. Computer Communications, 214, 100-112. https://doi.org/10.1016/j.comcom.2023.11.022.
55. Quan, M. K., Nguyen, D. C., Nguyen, V. D., Wijayasundara, M., Setunge, S., & Pathirana, P. N. (2024). HierSFL: Local differential privacy-aided split federated learning in mobile edge computing. arXiv:2401.08723 [Cryptography and Security]. https://doi.org/10.48550/arXiv.2401.08723.

Biographies

1. **LiPing He** Master Degree, Senior Engineer. Her current main research directions include Artificial Intelligence, Big Data, and Computer Technology Research. She has published nearly 15 academic papers in domestic and international journals; presides over or participates in 12 research projects that have been approved and are in progress, and 2 projects that have been completed; in addition, she has compiled 1 textbook as the chief editor and 2 textbooks as the associate editor.

邊緣計算架構下學生學習態勢實時分析與個性化路徑的聯邦遷移學習算法研究

何麗萍[1]

[1]廣東司法警官職業學院，廣州，中國，510520

摘要：在教育數字化背景下，傳統基於雲端的學情分析存在實時性不足、數據隱私風險及跨域適配困難等挑戰。本研究提出三級算法體系，構建包含邊緣預處理、聯邦優化與遷移適配的協同架構：邊緣層實現多模態數據的輕量化特徵提取；聯邦層採用FedProx算法結合差分隱私與同態加密確保跨校協作安全；遷移層通過域對抗訓練與知識圖譜強化學習實現跨域特徵對齊與個性化學習路徑生成。實驗表明，該算法在邊緣端實現低延遲推斷，有效提升跨域推薦精度並降低隱私風險，為教育智能化提供實用技術方案。

關鍵詞：邊緣計算；學情實時分析；個性化學習路徑；聯邦遷移學習

1. 何麗萍，現任廣東司法警官職業學院信息管理系高級工程師。目前主要研究方向包括人工智能、大數據及計算機技術研究。她在國內外期刊已發表學術論文近15篇；主持或參與12項已立項並在研的科研項目，以及2項已結題項目；此外，她還主編了1部教材，副主編了2部教材。