Edge Computation Network Resource Scheduling and Privacy Protection Mechanism Based on Federated Learning

Zongjian Fu

School of Artificial Intelligence, Zhengzhou Railway Vocational & Technical College, Zhengzhou, 451460, China
E-mail: fuzongjian@163.com

Received 05 September 2025; Accepted 30 September 2025

Abstract

The diversity of edge devices and unpredictable network environments, along with the quick expansion of Internet of Things devices, have resulted in ineffective resource scheduling and privacy leakage threats. To address resource contention issues in dynamic environments, ensure security protection on resource-constrained edge devices, and improve the efficiency of resource scheduling and privacy protection in edge computing networks, this study designs a network computing resource optimization and data security protection solution based on federated learning. The research method deeply integrates federated learning with edge computing, adopts hierarchical federated learning technology to solve resource scheduling problems, and uses the Lagrange optimization method to achieve a closed-form solution for resource scheduling. By combining local differential privacy and homomorphic encryption technologies, a noise injection, masking mechanism, and ciphertext computation scheme are designed to ensure data privacy and security. The outcomes indicated that the task completion delay of the research

Journal of Cyber Security and Mobility, Vol. 14_6, 1321–1346.
doi: 10.13052/jcsm2245-1439.1462

method was 1250 ms when the data volume was 100 GB. Furthermore, when the privacy budget increased from 0.1 to 10, the efficiency value decreased by only 0.26. In practical application testing, the communication volume of the research method was 1.5MB per round when the number of clients was 100. Additionally, when the network fluctuation level increased from level 1 to level 5, the dropping rate increased by only 3.3%. The above results indicate that the resource scheduling and privacy protection mechanism based on federated learning in edge computing networks, as proposed by the research, is highly practical, robust and secure. It effectively solves the problems of low resource scheduling efficiency, poor adaptability and inadequate privacy protection capabilities in edge computing networks.

Keywords: Federated learning, edge computing network, Lagrange optimization, local differential privacy, homomorphic encryption.

Introduction

The amount of data produced by a vast number of end devices is increasing exponentially due to the quick development of artificial intelligence (AI) and the Internet of Things (IoT). However, a substantial quantity of bandwidth is used while sending so much raw data to the cloud. According to surveys, data transmission energy consumption alone accounts for more than 60% of the total energy consumption in industrial IoT scenarios [1]. These include the potential for reaction delays from remote processing in the cloud, which can directly result in safety accidents and make it challenging to meet the millisecond-level needs of real-time applications like industrial control, remote surgery, and autonomous driving [2]. Edge computing (EC) can reduce latency and reduce dependence on the cloud by shifting computing power to the network edge and processing data locally [3]. However, edge nodes have limited computing power, spectrum, and storage resources, which are unevenly distributed. It is necessary to dynamically schedule multi-user tasks to avoid task failure due to load imbalance. Furthermore, edge nodes are more susceptible to physical attacks. If local data is not encrypted or anonymized, it may be used to reverse-engineer user behavior patterns [4]. Federated learning (FL) adopts the paradigm of "data stays put, models move". After terminal devices complete model training locally, they only transmit model weight parameters to edge servers (ESs) to participate in

global aggregation, which can effectively prevent the leakage of raw data [5]. To achieve the goal of efficient resource scheduling (RS) while ensuring privacy in EC networks, this study innovatively designed an EC resource optimization and data security protection solution based on FL. The study implements dynamic resource optimization through a three-layer architecture, defines local loss functions (LFs) and latency models, and combines the Lagrange method to optimize power allocation and offloading decisions. By integrating local differential privacy (LDP) and homomorphic encryption (HE) technologies, a "cloud-edge-end" end-to-end privacy protection (PP) system has been constructed. Through noise injection, secure aggregation, and dynamic privacy budget updates, the security of data during transmission and computation is ensured. Theoretical support for RS and PP in IoT devices across a range of scenarios is expected from this study methodology.

1 Related works

The reasonable allocation of EC network resources and PP can effectively improve resource utilization efficiency and ensure data privacy (DP) and security. Thus, the research on RS and PP mechanisms for EC networks is of great significance. In an effort to solve the issues of low privacy security and high time consumption in EC terminal RS, Zhan Z et al. suggested a privacy-preserving RS technique based on partially observable Markov decision processes and reinforcement learning. This study used privacy entropy to quantify task uncertainty and data security. The outcomes demonstrated that the proposed method achieved good data security and energy efficiency [6]. Kong L et al. addressed the issues of high data transmission latency and heavy cloud load caused by the surge in IoT devices, proposing the use of EC as a new decentralized computing model. According to the findings, the research approach improved the scalability of IoT systems, decreased network latency, and lessened the strain on the cloud [7]. Wang R et al. addressed privacy risks and efficiency issues in medical IoT collaborative training by proposing an edge-side PP method combining key sharding and parameter perturbation. The findings demonstrated the effectiveness and anti-attack capabilities of the research methodology [8]. McEnroe P et al. addressed the issue that cloud-based AI paradigms struggle to meet the low-latency, low-power requirements of drone IoT applications. They proposed running AI on devices or near the user edge to improve drone services. The findings

demonstrated that the study approach may offer thorough recommendations for maximizing drone performance [9]. Hua H et al. addressed the limitations of traditional non-AI methods in enhancing EC performance by proposing a machine learning-based approach to optimize EC performance. According to the findings, the research approach offered fresh perspectives on examining the advantageous connection between EC and AI [10].

Numerous academics from both local and foreign universities have studied and used FL technologies in great detail. Wen J et al. systematically reviewed the current state of FL research, focusing on bottlenecks such as privacy security and communication overhead. The study analyzed five dimensions and summarized practical application results. The results indicated that FL could effectively solve data silo problems [11]. Liu Y et al. proposed a unified framework for balancing PP and efficiency in vertical FL. The findings suggested that the study approach might successfully balance performance and privacy [12]. Zhu J et al. addressed the security and reliability concerns of centralized architectures in FL by proposing a blockchain-enabled authorized FL solution. The findings suggested that the research approach might successfully improve FL's decentralized security features [13]. Liu J et al. conducted a systematic review of FL technology to address the challenge of machine learning with distributed data that cannot be directly shared. The results showed that FL can effectively achieve collaborative modeling under data security [14]. Ye M et al. systematically reviewed the research challenges and latest developments in heterogeneous FL, addressing the statistical, modeling, communication, and device heterogeneity issues faced by FL in industrial applications. The results indicated that heterogeneous FL could effectively address the heterogeneity challenges in practical applications [15].

In summary, the current research has shown notable advancements in EC resource scheduling and PP. However, there are still challenges, including suboptimal dynamic adaptability and inadequate heterogeneous support. Additionally, there are concerns such as centralized privacy risk, ineffective privacy efficiency trade-offs, and an absence of multi-level PP. Each node may train local models using its own data thanks to FL technology. Then, the adjusted model parameters are transmitted to the main node for global integration, achieving data PP while completing model training. Therefore, this study designs a network computing resource optimization and data security protection scheme based on FL. It is expected to meet the requirements of mechanism design and achieve efficient RS while protecting privacy.

2 RS and PP Mechanisms in EC Networks

2.1 IoT Device Association and RS Mechanism Based on Hierarchical FL

With the development of IoT, autonomous driving, and other applications, traditional edge network RS methods face challenges such as difficulty in adapting to dynamic environments, poor scalability, and a lack of intelligent decision-making capabilities [16]. Device association and RS can be integrated using hierarchical FL technology, which can dynamically modify FL frequency and participating device selection to accommodate resource limitations and network conditions [17]. Therefore, in response to the shortcomings of existing edge resource scheduling methods in terms of dynamic adaptability, heterogeneity support, and cross layer collaboration, a resource allocation mechanism based on hierarchical federated learning is proposed. The closed-form solution for power and offloading decisions is achieved using the Lagrange optimization method. This method improves adaptability to dynamic network environments and increases the efficiency of resource utilization. The scenario diagram of hierarchical federated learning in the IoT is shown in Figure 1.

In Figure 1, the FL architecture in IoT has three layers, including cloud servers, EC nodes, and heterogeneous terminal devices. Data is uploaded from edge models to EC servers for local aggregation, then uploaded to the cloud for global optimization. The optimized model is distributed level by level through the "global model download" path. To avoid uploading raw data to the server, the study defines a local LF under the computing network device framework, as shown in Equation (1).

$$F_k(\omega_k) = \frac{1}{D_k} \sum_{i \in \mathcal{D}_k} f_i(\omega_k) \tag{1}$$

In Equation (1), $F_k(\omega_k)$ displays the local LF value of device k. ω_k is the local model parameter. D_k is the data size. $\mathcal{D}_k$ is the dataset. The study establishes a delay model for the data offloading phase in an attempt to overcome the problems of inadequate processing power and resource limitations in industrial IoT devices, as shown in Equation (2).

$$\begin{cases} T_n^o = \max\limits_{k \in \mathcal{G}_n} t_{k,n}^o \\ t_{k,n}^o = \dfrac{\alpha_k D_k}{r_k} = \dfrac{\alpha_k D_k}{\frac{B_n}{|\mathcal{G}_n|} \log_2(1 + \frac{h_{k,n} p_k}{N_o})} \end{cases} \tag{2}$$

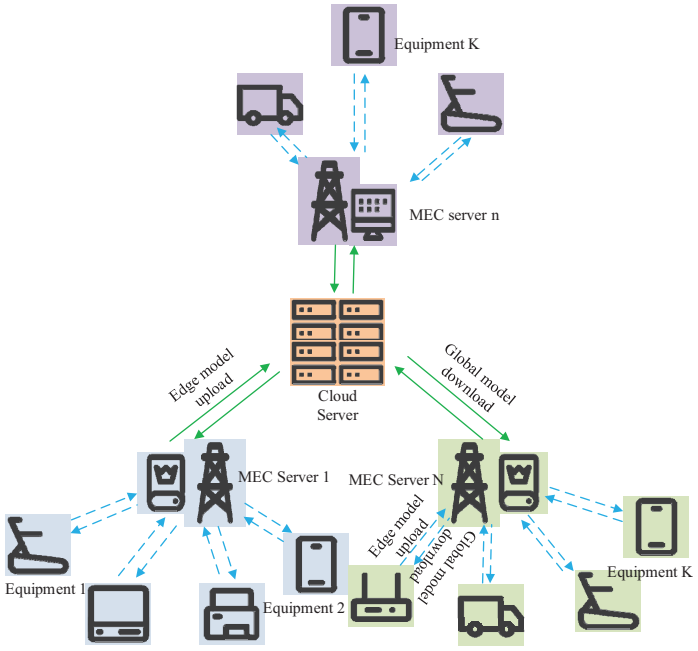


Figure 1 Hierarchical FL scenario in the IoT (Icons in the picture are sourced from: https://iconpark.oceanengine.com/home).

In Equation (2), T_n^o is the offloading delay of ES n. $t_{k,n}^o$ is the delay of offloaded data. α_k is the offloading ratio. B_n is the total bandwidth provided. $\mathcal{G}_n$ is the quantity of associated devices. $h_{k,n}$ is the channel gain. p_k is the transmission power (TP). N_o displays the background noise power. The study quantifies the communication efficiency of FL enabled by EC to meet the real-time requirements of IoT, as shown in Equation (3).

$$t_{k,n}^u = \frac{d_k}{r_k} = \frac{d_k}{\frac{B_n}{|\mathcal{G}_n|}\log_2(1+\frac{h_{k,n}p_k}{N_0})} \tag{3}$$

In Equation (3), $t_{k,n}^u$ is the delay for uploading model parameters. d_k is the data size of model parameters ω_k. d_k is the uplink transmission rate. Next, Equation (4) illustrates how the study creates a new edge model by performing weighted aggregation of the model parameters trained locally by all devices connected to the ES and the model parameters trained by the ES itself.

$$\omega_n^e = \frac{\sum_{k\in\mathcal{K}_n}(1-\alpha_k)D_k\omega_k + O_n\omega_n}{D_n} \tag{4}$$

In Equation (4), ω_n^e represents the model parameters after aggregation. $\mathcal{K}_n$ represents the device set. O_n represents the total amount of offloaded data. ω_n represents the model parameters trained locally. D_n represents the total data volume of associated devices. To reflect the impact of device heterogeneity in the IoT RS process, the study explicitly identifies system bottlenecks, as shown in Equation (5).

$$T_n^e = Y(\zeta, \theta) \max_{k \in \mathcal{G}_n} (t_{k,n}^l + t_{k,n}^u) \tag{5}$$

In Equation (5), T_n^e represents the perception layer (PL) latency. $Y(\zeta, \theta)$ denotes the quantity of edge iterations required to achieve edge accuracy ζ. θ represents the target accuracy for local model training. $t_{k,n}^l$ denotes the local model training latency. In IoT systems, by quantifying and analyzing the latency within the IoT system, the design and performance of the IoT RS module can be optimized. The IoT system network latency framework is illustrated in Figure 2.

In Figure 2, the total delay of the IoT system network includes the PL, network layer, and application layer. Among them, the PL covers data acquisition, analog-to-digital conversion, and packaging delay. The network

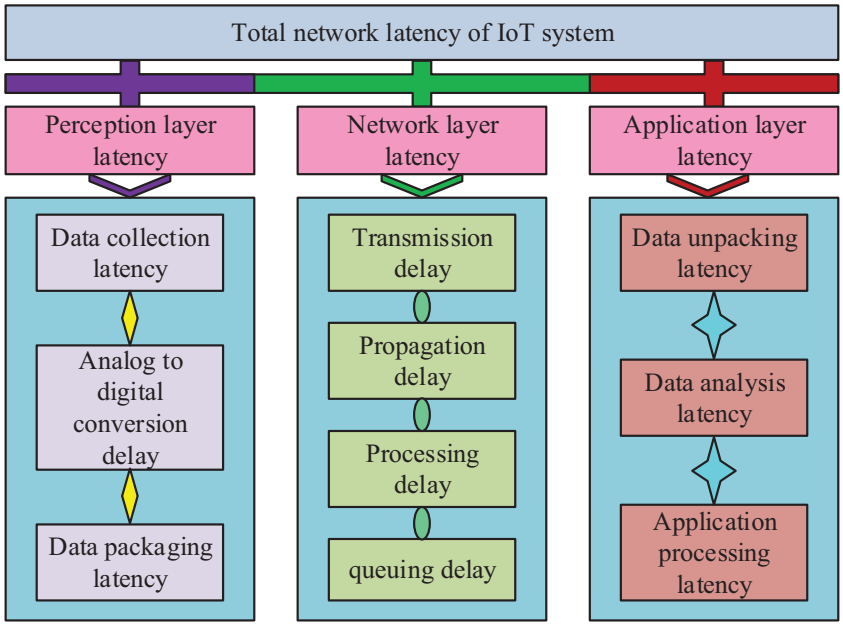


Figure 2 Network latency framework of IoT system.

layer includes transmission, propagation, processing, and queuing delay. The application layer involves unpacking, analysis, and application processing delay. To ensure real-time constraints in IoT scenarios, the study introduces a quantitative model of the total system delay, as shown in Equation (6).

$$T^{total} = \max_{n \in \mathcal{N}} (T_n^o + T_n^e + T_{n,S}^u) \tag{6}$$

In Equation (6), T^{total} displays the total system delay. T_n^o is the network layer delay. $T_{n,S}^u$ is the application layer delay. $\mathcal{N}$ is the ES cluster. The study use the Lagrange method to transform the limited optimization problem into a solvable unconstrained form in order to solve the TP allocation problem, as shown in Equation (7).

$$\begin{aligned} \mathcal{F}(\mathbf{p}, \mathbf{W_1}, \lambda, \varphi) = W_1 + \sum_{k \in \mathcal{G}_n} \lambda_k \left[\frac{\alpha_k D_k + d_k}{\frac{B_n}{|\mathcal{G}_n|} \log_2 \left(1 + \frac{h_{k,n} p_k}{N_0}\right)} - W_1 \right] \\ + \varphi \left(\sum_{k \in \mathcal{G}_n} p_k - P_n \right) \end{aligned} \tag{7}$$

In Equation (7), $\mathcal{F}(\cdot)$ is the Lagrange function. $\mathbf{p}$ is the power transmission vector. $\mathbf{W_1}$ is the auxiliary variable. λ is the Lagrange multiplier. φ is the total power constraint multiplier. Next, to balance local computation and offloading overhead, the study introduces a closed-form solution for power allocation and offloading ratio decision, as shown in Equation (8).

$$\begin{cases} W_1^*(\alpha_k) = \dfrac{\sqrt{b^2 + 4aP_n} + b}{2P_n} \\ \alpha_k^* = \dfrac{c \frac{h_{k,n}}{N_0} \left[\nu_k \frac{C_k}{f_k} - f_2(\nu_k) \right] - d - e}{2P_n D_k |\mathcal{G}_n|^2 \ln^2 2} \end{cases} \tag{8}$$

In Equation (8), W_1^* represents the optimal value of the upper bound of transmission delay. a and b are the second-order and first-order coupling terms for channel quality and data volume, respectively. α_k^* denotes the optimal offloading ratio. ν_k is the Lagrange multiplier, and c is the joint cost term for power and bandwidth. d is the base transmission cost term. e is the fixed cost term for model transmission. In summary, the IoT device association and RS process based on hierarchical FL is shown in Figure 3.

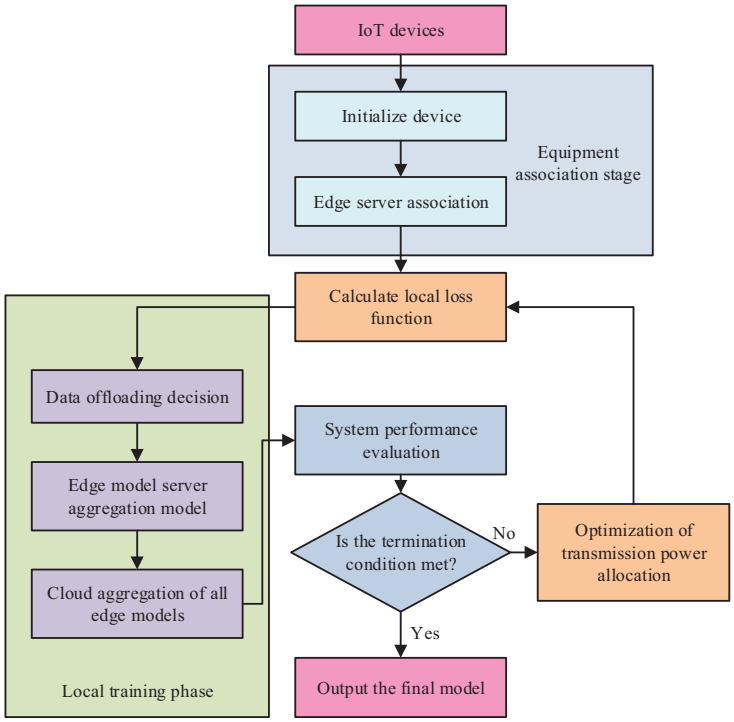


Figure 3 Collaborative and RS scheme for IoT terminals based on hierarchical FL.

In Figure 3, the core component of the IoT device association and RS mechanism based on hierarchical FL is the local training phase. Devices calculate local LFs and trigger TP optimization allocation to save energy and improve efficiency. Meanwhile, data offloading decisions determine the balance between local computing and edge collaboration. The locally updated models generated by training are fused with system performance evaluation and resource optimization status during the hierarchical aggregation process within the architecture to jointly determine whether the termination conditions are met. If not met, the optimized model and RS strategy drive the next round of iterative training. If met, the final optimized global model is output.

2.2 Federated Learning Multi Level Data Protection Method Based on FSS-TDG Algorithm

The research design of a collaborative and resource optimization scheme for terminal devices based on hierarchical FL can effectively enhance the

environmental adaptability and scalability of edge network RS systems. However, due to the frequent flow of data between distributed edge nodes and the involvement of multiple parties, the risk of privacy leakage is significantly increased [18]. LDP can add noise to data locally to protect DP while ensuring the accuracy of model training [19]. In response to the issues of centralization risk, low efficiency, and a single protection link in existing PP methods, a three-level "cloud edge end" PP system was constructed. This system integrates LDP and HE to realize full-chain ciphertext processing and noise injection from data collection and transmission to model aggregation. It effectively resists various privacy attacks while ensuring model accuracy. To ensure controllable privacy strength, Gaussian noise is added to the model parameters. The specific algorithm formula is shown in Equation (9).

$$\tilde{\theta}_i^{(t)} = \theta_i^{(t)} + G(0, \sigma_i^2 \cdot \Delta f_i \cdot \rho_i^{(t)}) \tag{9}$$

In Equation (9), $\tilde{\theta}_i^{(t)} = \theta_i^{(t)} + G(0, \sigma_i^2 \cdot \Delta f_i \cdot \rho_i^{(t)})$ is the parameter of the i-th client after the t-th round of perturbation. $\theta_i^{(t)}$ is the original parameter after local training. G is the Gaussian noise distribution. Δf_i is the model gradient sensitivity. $\rho_i^{(t)}$ is the local privacy budget. Next, the aggregation of the parameters of the territorial clients is studied, as shown in Equation (10).

$$\begin{cases} \hat{\theta}_n^{(t)} = \sum\limits_{i \in \mathcal{C}_n} \tilde{\theta}_i^{(t)} \oplus \mathbf{M}_{n,i}^{(t)}, \\ \mathbf{M}_{n,i}^{(t)} = \mathrm{PRG}(\mathrm{SS}_{n,i}^{(t)}) \end{cases} \tag{10}$$

In Equation (10), $\hat{\theta}_n^{(t)}$ is the mask parameter after aggregation. $\mathcal{C}_c$ is the client set. $\oplus$ is element-wise addition. $\mathbf{M}_{c,i}^{(t)}$ is the dynamic mask vector. $\mathrm{PRG}(\cdot)$ is the pseudorandom generator. $\mathrm{SS}_{c,i}^{(t)}$ is the secret seed. To further reduce the impact of untrusted edge nodes, the study introduced a global perturbation aggregation and privacy amplification model, as shown in Equation (11).

$$\bar{\theta}^{(t)} = \frac{1}{|\mathcal{E}|} \sum_{n \in \mathcal{E}} \hat{\theta}_n^{(t)} + \mathcal{L}\left(0, \frac{\beta \cdot \Delta F}{\epsilon_g}\right) \tag{11}$$

In Equation (11), $\bar{\theta}^{(t)}$ is the parameters aggregated globally. $\mathcal{E}$ denotes the set of ESs. $\mathcal{L}(\cdot)$ refers to the Laplace distribution. ϵ_g signifies the global privacy budget. ΔF denotes the global sensitivity. β represents the trustworthiness weights of ESs. Finally, the study dynamically adjusts the budget for

the next round using a dynamic update formula for the privacy budget, as displayed in Equation (12).

$$\begin{cases} \rho_i^{(t+1)} = \rho_i^{(t)} \cdot e^{-\gamma \cdot \|\theta_i^{(t)} - \bar{\theta}^{(t)}\|_2} \\ \epsilon_g^{(t+1)} = \epsilon_{total} - \sum_i \rho_i^{(t)} \end{cases} \tag{12}$$

In Equation (12), $\rho_i^{(t+1)}$ represents the budget for the next round. γ is the decay coefficient. ϵ_{total} is the total privacy budget of the system. HE can perform computations on encrypted data, and when decrypted, the results are consistent with those obtained from computations performed on plaintext. This protects DP during transmission and computation [20]. Subsequently, to ensure the confidentiality of information during interaction and processing, a PP scheme based on HE is designed for FL. In Figure 4, the particular model is displayed.

In Figure 4, a multi-party collaborative learning system consisting of clients, ESs, model aggregation centers, and key management nodes is

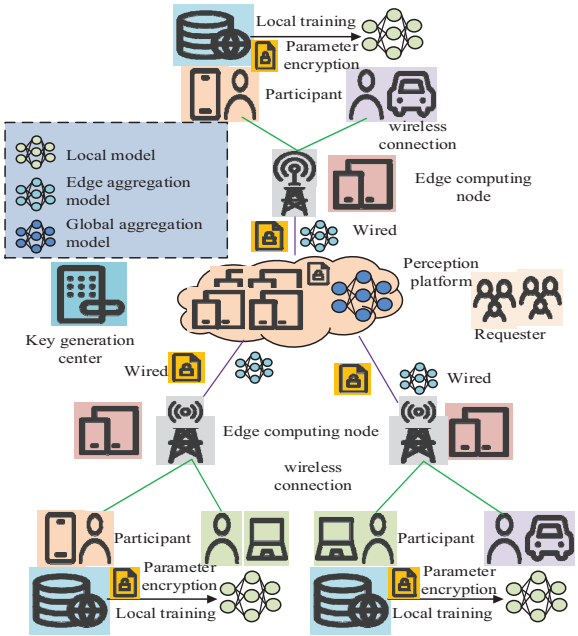


Figure 4 Collaborative learning PP mechanism integrating HE (Icons in the picture are sourced from: https://iconpark.oceanengine.com/home).

designed based on the HE FL framework. It achieves end-to-end encrypted computation through wireless and wired hierarchical transmission. To ensure that even if the ES is compromised, the attacker cannot decrypt or reverse engineer the original parameters, the study introduces participant-local encryption and model perturbation, as shown in Equation (13).

$$\mathbf{c}_i^{(t)} = \mathrm{Enc}_{\mathrm{pk}}(\theta_i^{(t)} + \mathcal{L}(0, \sigma_i^2)) \oplus \mathbf{m}_i \tag{13}$$

In Equation (13), $\mathbf{c}_i^{(t)}$ is the encrypted perturbation parameter uploaded. $\mathrm{Enc}_{\mathrm{pk}}(\cdot)$ is the Paillier addition HE function using the public key pk. $\mathbf{m}_i$ is a one-time random mask. To achieve noise layered injection, the study performs weighted aggregation of the encryption parameters of the jurisdiction client, as shown in Equation (14).

$$\widetilde{\mathbf{c}}_n^{(t)} = \sum_{i \in \mathcal{C}_n} \mathbf{c}_i^{(t)} \odot \mathbf{k}_n + \mathcal{L}(0, \sigma_n^2) \tag{14}$$

In Equation (14), $\widetilde{\mathbf{c}}_n^{(t)}$ display the re-encryption parameter. $\odot$ is the homomorphic scalar multiplication. $\mathbf{k}_n$ is the private key. Finally, the study introduces verification of the obtained global model parameters, as shown in Equation (15).

$$\begin{cases} \hat{\theta}_{\mathrm{global}}^{(t)} = \mathrm{Dec}_{\mathrm{sk}_n}(\mathbf{C}_{\mathrm{global}}^{(t)}) - \mathbf{m}_{\mathrm{total}} \\ \mathrm{verify}(\|\hat{\theta}_{\mathrm{global}}^{(t)} - \theta_{\mathrm{prev}}\|_2 \leq \tau) \\ \mathbf{C}_{\mathrm{global}}^{(t)} = \prod_{n=1}^{E} (\widetilde{\mathbf{c}}_n^{(t)})^{\alpha_n} \bmod N^2 \end{cases} \tag{15}$$

In Equation (15), $\hat{\theta}_{\mathrm{global}}^{(t)}$ represents the global model parameters after decryption and demasking. $\mathrm{Dec}_{\mathrm{sk}_n}(\cdot)$ denotes the Paillier decryption function using the edge private key sk_n. $\mathbf{C}_{\mathrm{global}}^{(t)}$ represents the encrypted parameters of the global aggregation. θ_{prev} denotes the global model parameters from the previous round. τ denotes the anomaly detection threshold. E is the total quantity of ESs. The operational flow of the FL data protection mechanism proposed in this study, which combines LDP and HE technologies, is illustrated in Figure 5.

In Figure 5, a "cloud-edge-end" three-layer PP FL system is constructed by combining LDP and HE technologies to form a FL data protection mechanism. To create encrypted parameters, the client uses local data perturbation,

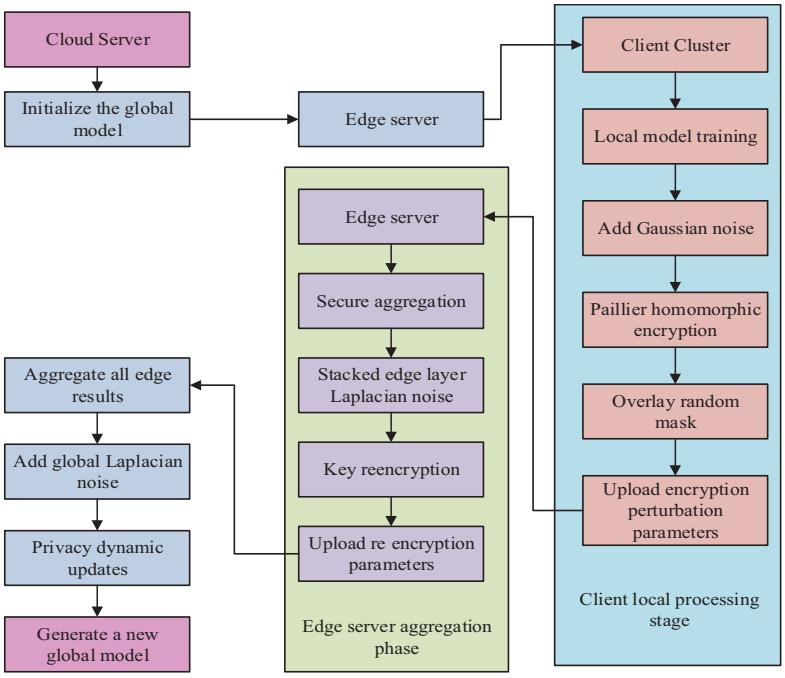


Figure 5 FL data protection mechanism combining LDP and HE technologies.

which it then uploads to the ES. The edge layer adds Laplace noise and re-encrypts the parameters, then securely aggregates them and reports them to the cloud. The cloud server integrates all edge data, injects global Laplace noise, and dynamically updates the privacy policy to generate a new global model. Overall, the RS and PP mechanism framework for EC networks based on FL is displayed in Figure 6.

In Figure 6, the RS and PP mechanism framework of the EC network based on FL constructs a dual-system architecture that deeply couples FL and EC. The FL side achieves PP through three-level collaboration between the cloud, edge, and endpoint. The client uses LDP noise injection and HE technology to ensure security, while the edge layer first isolates sensitive data and completes global privacy enhancement and model distribution in the cloud center. The EC side generates closed-loop offloading decisions based on real-time network information and optimizes task execution through resource virtualization and dynamic scheduling. The two systems work closely together to ultimately achieve unified PP, efficient resource utilization, and low-latency services.

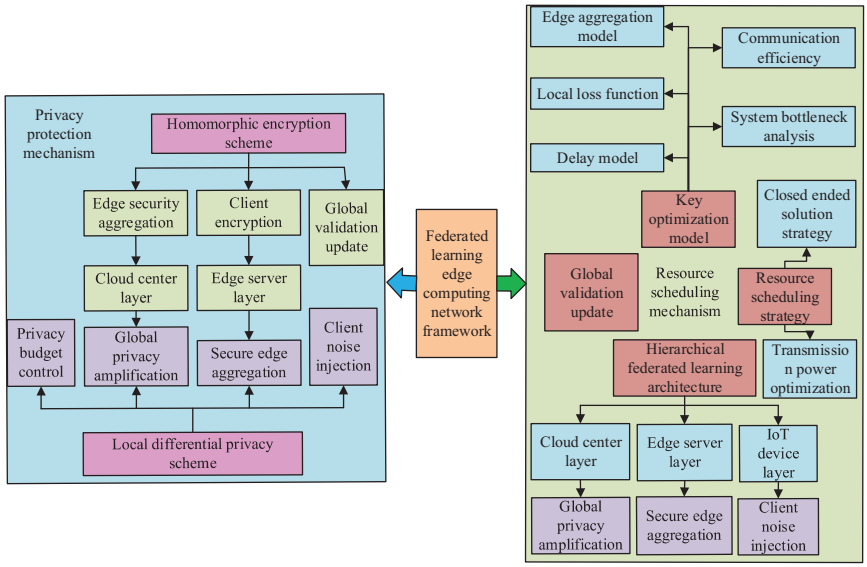


Figure 6 Framework of EC network RS and PP mechanism based on FL.

The proposed research on resource allocation and privacy protection mechanisms based on hierarchical federated learning significantly differs from existing solutions in the following aspects: (1) Deep integration of hierarchical federated learning and resource scheduling. Most existing studies adopt a flat FL architecture, which struggles to adapt to the heterogeneity and dynamics of edge networks. (2) Multi-level privacy protection mechanism combining LDP and HE. Most current approaches employ only a single privacy protection technique, making it difficult to maintain model accuracy while achieving end-to-end privacy protection. (3) Dynamic privacy budget and resource state linkage mechanism. The study introduces a dynamic privacy budget update strategy that can adjust the intensity of privacy protection based on network conditions, device trustworthiness, and task requirements, avoiding the efficiency loss or privacy leakage risks caused by fixed privacy budgets in traditional methods. In summary, this research not only achieves deep coupling between FL and EC in architectural design but also proposes multiple innovations at the algorithmic level, significantly enhancing the system's practicality, security, and robustness in dynamic environments.

Ranging from local training on client devices to global aggregation in the cloud, the entire research process can be divided into three stages: (1) Local training and preliminary protection on client devices. (2) ES aggregation

and secondary protection. (3) Global aggregation in the cloud with dynamic privacy management. To further enhance the efficiency and adaptability of resource allocation, the study introduces the FSS-TDG framework, combining genetic algorithm (GA) and particle swarm optimization (PSO) for joint optimization. The specific implementation is as follows: (1) FSS-TDG generates task-driven strategies. (2) GA/PSO optimizes strategies, where GA encodes strategies as chromosomes and PSO treats strategies as particle positions. (3) Parallel evaluation and feedback.

3 Verification of Network RS and PP Mechanisms Based on FL

3.1 Performance Testing of Network RS and PP Mechanisms

To verify the performance of the EC network's resource allocation and PP mechanisms based on federated learning, a simulation model is studied and constructed. Its experimental environment and specific configuration are shown in Table 1.

Moreover, to further ensure the reproducibility of the experiment, Tables 2 and 3 show the simulated environment parameters and the hyperparameter settings of the federated learning, respectively.

Tables 1, 2, and 3 present the results of performance testing conducted in the testing environment, with the experimental configuration, simulation environment parameters, and federated learning hyperparameter settings listed in the tables. This testing used the NIST+CIFAR-10 combined dataset. Compare these research methods with those of virtual network embedding (VNE), full duplex relay (FDR), and improved genetic algorithm (IGA). The specific configurations of each method are shown in Table 4.

In Table 4, three comparison methods are used to compare the specific configurations and research methods in the table. The task completion delays

Table 1 Test environment and specific configuration

Testing Environment	Specific Configuration
CPU	AMD Ryzen 9 7950X
GPU	NVIDIA RTX 4090
Storage	2TB NVMe SSD
Edge simulator	OMNeT++/INET
FL framework	PyTorch+Flower
Monitoring and analysis	Prometheus+Grafana

Table 2 Details of simulated environment parameters

Parameter Type	Parameter Name	Parameter Value/Range
Dataset	Name	NIST + CIFAR-10
	Total data volume	10–100 GB
	Client data distribution	Non-IID
Network topology	The number of edge nodes	10-20
	The number of clients	50-600
	Bandwidth range	10–100 Mbps
	Network latency	5–50 ms
Equipment configuration	Computing power	1–4 cores per device
	Storage capacity	1–8 GB RAM per device
	Power consumption model	Linear model (Joule/round)

Table 3 Hyperparameter settings for federated learning

Hyperparameter	Value
Learning rate	0.01
Number of local iterations	5
Global communication rounds	100
The client selects the proportion	0.5
Local privacy budget	0.1–10
Global privacy budget	1.0
The mask generates the seed	Randomly generated

Table 4 Specific configuration of benchmark methods

Method Name	Implementation Tools/Libraries	Key Parameter Settings
VNE	OMNeT++/INET	Virtual network mapping algorithm: Greedy strategy
FDR	Python/Custom	Full-duplex relay power control: Dynamic adjustment
IGA	Python/DEAP	Population size: 50, number of iterations: 100, crossover rate: 0.8

and task scheduling success rates of the four methods are compared under different data volumes and concurrent task counts. The results are shown in Figure 7.

In Figure 7(a), the task completion delay of all four methods increases with the increase in data volume. Among them, the increase trend of the proposed approach is the most gradual. When the data volume is 100GB, the task completion delay of the proposed approach is only 1250 ms. The increase trends of the other three methods are significantly greater than that of the proposed approach. In Figure 7(b), the TSSR of the proposed approach

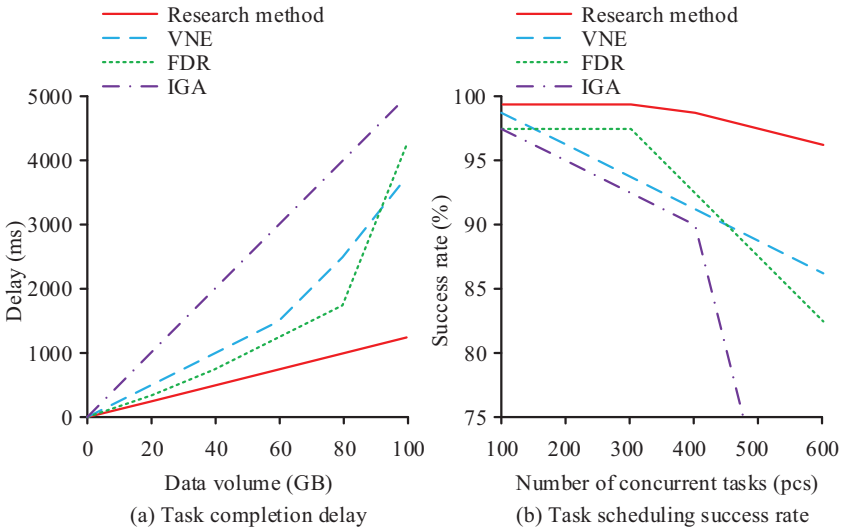


Figure 7 Delay in task completion and success rate of task scheduling.

remains stable at 98.2% when the number of concurrent tasks is less than 300. When the concurrent task increases to 600, the TSSR of the proposed approach is 96.5%, a decrease of only 1.7%. In contrast, the TSSRs of the other three methods are significantly lower than that of the proposed approach across all task counts. All things considered, the suggested method is more reliable and practical than the comparison method. The efficiency values of the four methods under different privacy budgets and the distribution of update loss rates within the network packet loss rate range are compared. Figure 8 displays the findings.

In Figure 8(a), the efficiency threshold of the system's FL under different privacy budgets is 0.5. The overall efficiency value of the proposed approach's FL efficiency exceeds the system efficiency threshold. The efficiency value of the suggested method drops from 0.98 to 0.72, a fall of 0.26, when the privacy budget is increased from 0.1 to 10. The FL efficiency of the other three approaches decreases far more than that of the suggested strategy when the privacy budget rises. In Figure 8(b), the system update loss rate threshold is 9% in different network packet loss rate intervals. The overall distribution of the update loss rate of the proposed approach is below the loss rate threshold in different network packet loss rate intervals. However, the update loss rates of the other three methods all exceeded the system loss rate threshold as the network packet loss rate increased. Overall, the

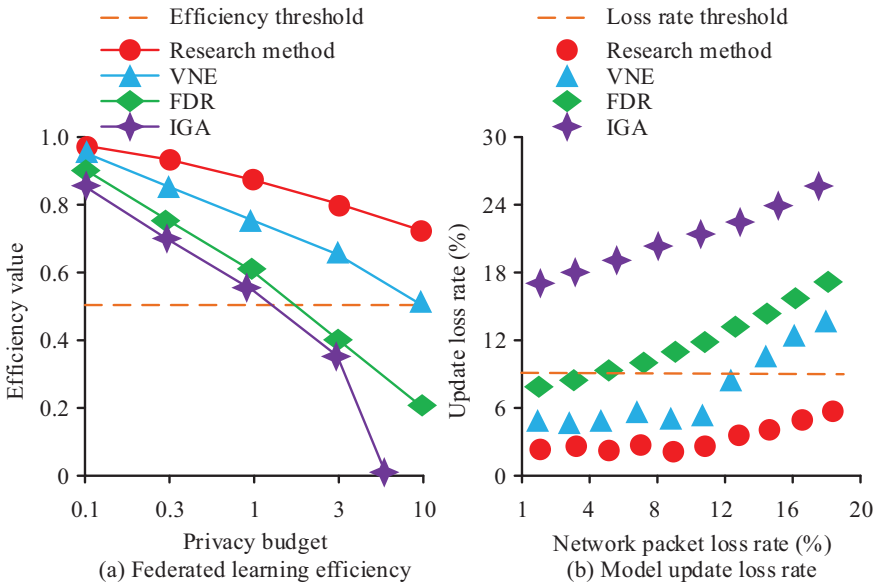


Figure 8 FL efficiency and model update loss rate.

research approach is more effective and resilient to network changes than the comparison method. Comprehensive analysis shows that the RS and PP mechanisms based on FL EC networks proposed in this study have good practicality, reliability, efficiency, and robustness.

3.2 The Practical Application Effects of Network RS and PP Mechanisms

Based on the verified performance of the RS and PP mechanisms of the FL-based EC network, the study further verifies the practical application effects of the proposed approach. The study uses the CityScapes dataset and builds the FedEdge-Industrial Validation Platform experimental platform. The proposed approach is compared with three other methods: VNE, FDR, and IGA. The four techniques' variations in communication bandwidth occupancy and model classification accuracy are examined as the number of training repetitions rises. The results are shown in Figure 9.

In Figure 9(a), the model classification accuracy and communication bandwidth occupancy rate of the research method quickly converges to stable values in the 60th round of training. The model classification accuracy and communication bandwidth occupancy rate are stable at 99.5% and 10.2%,

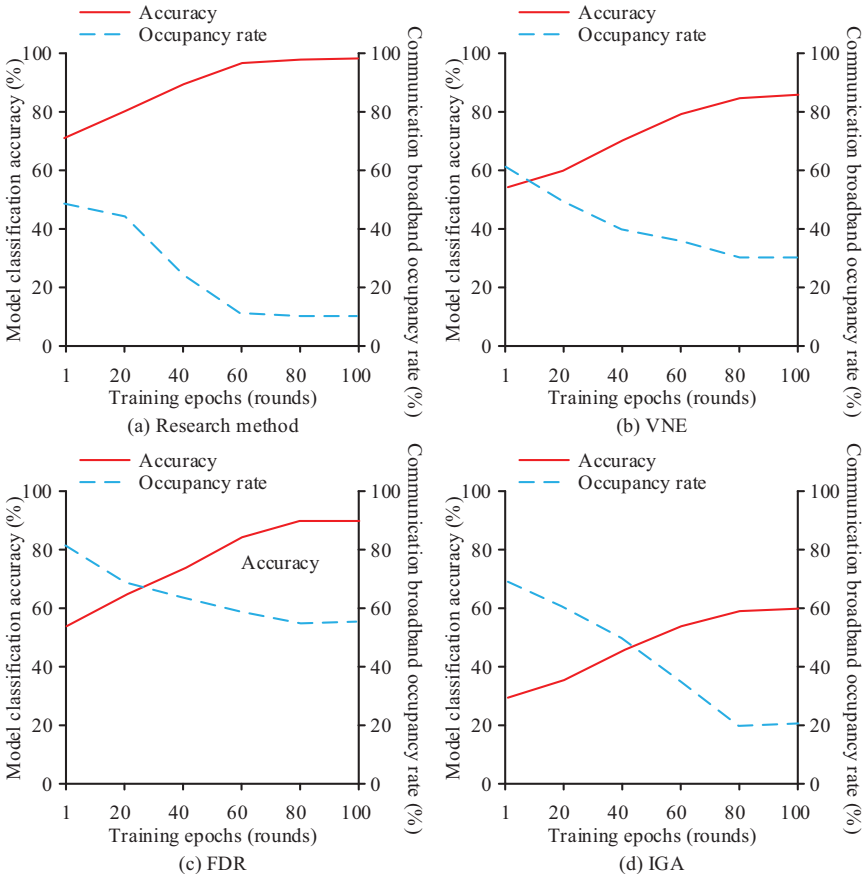
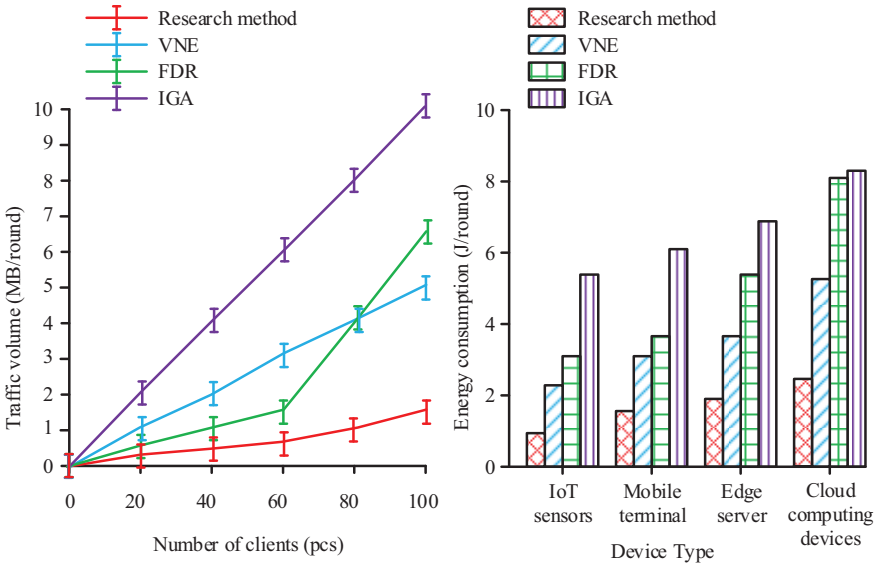


Figure 9 Model classification accuracy and communication bandwidth occupancy under different training epochs.

respectively. This conforms to the ideal trend of "high accuracy, low bandwidth". In Figure 9(b), the model classification accuracy and communication bandwidth occupancy of the VNE method only fully converge and reach stable values in the 80th round of training. The stable values of the model classification accuracy and communication bandwidth occupancy are 84.7% and 30.6%, respectively. Figure 9(c) shows that the model classification accuracy and communication bandwidth occupancy of the FDR method only fully converge and reach stable values in the 80th round of training. However, its communication bandwidth occupancy does not meet the low-bandwidth requirement. Figure 9(d) shows that the IGA method's model classification

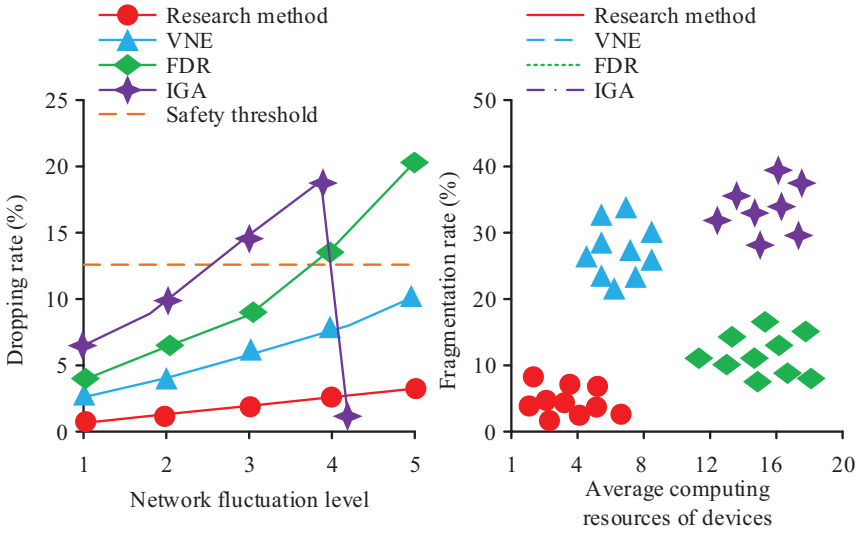


(a) Ability to resist scale expansion (b) Energy consumption of edge devices

Figure 10 The ability of the model to resist the scale expansion and energy consumption of edge devices.

accuracy and communication bandwidth occupancy fully converged in the eighth round of training. However, its model classification accuracy does not meet the high-accuracy requirements. Overall, the suggested strategy has superior collaborative optimization capabilities when compared to the comparable methodologies. The communication volume of the four methods under different numbers of clients and the energy consumption under different device types are compared. Figure 10 presents the findings.

As the client rises, Figure 10(a) shows that the communication volume of the four ways progressively grows. The smallest growth among them is in the research approach. The proposed approach's communication volume is only 1.5 MB every round when there are 100 clients. As the client increases, the increase in communication volume of the other three methods is significantly greater than that of the proposed approach. In Figure 10(b), the energy consumption of the proposed approach is 0.9 J/cycle on IoT sensors, 1.5 J/cycle on mobile terminals, 1.9 J/cycle on ESs, and 2.3 J/cycle on traditional cloud computing devices. Compared to the suggested way, the other three approaches' energy consumption on various device kinds is noticeably higher. In summary, the suggested method is more efficient and



(a) Equipment disconnection rate (b) Calculate resource fragmentation rate

Figure 11 Device dropout rate and computing resource fragmentation rate.

scalable than the comparison method. The dropping rate of the four methods at different network fluctuation levels, as well as the distribution of fragmentation rate and average computing resources of devices, are compared. Figure 11 presents the findings.

In Figure 11(a), the system's dropping rate safety threshold under different network fluctuation levels is 12.5%. The dropping rate of the proposed approach increases as the network fluctuation level increases. When the network fluctuation level is Level 1, the dropping rate of the proposed approach is 0.2%. When the network fluctuation level is Level 5, the dropping rate of the proposed approach is 3.5%. At various network fluctuation levels, the dropping rates of the other three approaches are noticeably higher than the suggested method's. In Figure 11(b), the fragmentation rate of the proposed approach is concentrated below 10%, and the average computing resources of devices are distributed within 8 cores, satisfying the ideal distribution of "low core count-low fragmentation rate". However, the fragmentation rate and average computing resources of devices for the other three methods clearly do not satisfy the distribution of "low core count-low fragmentation rate". Overall, the suggested strategy uses more resources and is more resilient to network fluctuations than the comparison method. In summary, the RS and PP mechanisms based on FL in EC networks proposed in this study have

good collaborative optimization capabilities, scalability, efficiency, and reliability. Overall, the study solves the dynamic and heterogeneous problems in resource scheduling through hierarchical federated learning and Lagrangian optimization. Efficient and end-to-end PP has been achieved through the dual privacy mechanism of LDP+HE. The experimental results demonstrate that the research method outperforms traditional methods in terms of task delay, communication overhead, privacy budget sensitivity, and network fluctuation tolerance. This indicates its high practicability, robustness, and security in actual EC scenarios.

4 Conclusion

Massive volumes of data have been produced in scenarios like intelligent transportation and industrial IoT due to the rapid expansion of IoT devices worldwide. This research creatively suggested an RS and PP mechanism based on a FL EC network to enhance resource usage and strike a compromise between privacy and efficiency. The proposed approach established a dual-system architecture that deeply integrated FL and EC. The client side employed LDP noise injection and HE technology to ensure data security, while resource virtualization and dynamic scheduling were utilized to optimize task execution. The outcomes indicated that the TSSR of the proposed approach remained stable at 98.2% when the concurrent task was less than 300. When the concurrent task increased to 600, the TSSR of the proposed approach was 96.5%, a decrease of only 1.7%. The update loss rate of the proposed approach was generally distributed below the loss rate threshold in different network packet loss rate intervals. In practical application testing, the model classification accuracy and communication bandwidth occupancy of the proposed approach both reached stable values of 99.5% and 10.2%, respectively, in the 60th round. The energy consumption of the proposed approach was 0.9 J/round on IoT sensors, 1.5 J/round on mobile terminals, 1.9 J/round on ESs, and 2.3 J/round on traditional cloud computing devices. Overall, the research's solution for network computing resource optimization and data security protection based on FL has good scalability and reliability. However, the research test uses a highly controlled, ideal simulation environment. Subsequent research can develop dynamic simulation tools to simulate extreme network fluctuations and enhance the proposed approach's comprehensiveness. In addition, the FSS-TDG algorithm proposed in this study shows significant advantages in EC resource allocation and PP. The key findings are as follows: In a dynamic network environment, the algorithm

achieves a closed-form solution for resource allocation through hierarchical federated learning and Lagrange optimization. This significantly reduces task completion delay and communication overhead. The multi-level PP mechanism combining LDP and HE can maintain high model utility even with a privacy budget of 10, demonstrating good privacy utility trade-off ability. This algorithm exhibits excellent scalability and robustness in practical applications, especially when it comes to maintaining stable performance as the number of clients increases or network fluctuations intensify. The extensive impact of the FSS-TDG algorithm on this field is primarily evident in its ability to provide a scalable, low-latency, integrated solution for resource scheduling and PP for heterogeneous edge devices. It has promoted the actual implementation of federated learning in EC, especially in scenarios with high real-time requirements such as industrial IoT and intelligent transportation. This provides a methodological paradigm combining optimization theory and cryptography for subsequent interdisciplinary research. It has high theoretical value and practical significance.

References

[1] Lata M, Kumar V. Security and privacy issues in fog computing environment. International Journal of Electronic Security and Digital Forensics, 2022, 14(3): 289–307. DOI:10.1504/ijesdf.2022.122588.

[2] Singh A, Satapathy S C, Roy A, Gutub A. Ai-based mobile edge computing for iot: Applications, challenges, and future scope. Arabian Journal for Science and Engineering, 2022, 47(8): 9801–9831. DOI:10.1007/s13369-021-06348-2.

[3] Nguyen D C, Pham Q V, Pathirana P N, Pathirana P N, Ding M, Seneviratne A, et al. Federated learning for smart healthcare: A survey. ACM Computing Surveys (Csur), 2022, 55(3): 1–37. DOI:10.1145/3501296.

[4] Hasanvand M, Nooshyar M, Moharamkhani E, Selyari A. Machine learning methodology for identifying vehicles using image processing//Artificial Intelligence and Applications. 2023, 1(3): 170–178. DOI:10.47852/bonviewAIA3202833.

[5] Hebbi C, Mamatha H. Comprehensive Dataset Building and Recognition of Isolated Handwritten Kannada Characters Using Machine Learning Models. Artificial Intelligence and Applications, 2023, 1(3):179–190. DOI:10.47852/bonviewAIA3202624.

[6] Zhan, Z., Wang, X., Liu, Y., Sun, Z., and Gu, C. Integration and Optimization Strategy of Blockchain-Enabled Edge Computing System

for Internet of Vehicles. Journal of Cyber Security and Mobility, 2025 14(02), 391–432. https://doi.org/10.13052/jcsm2245-1439.1426.

[7] Kong L, Tan J, Huang J, Chen G, Wang S, Jin X, et al. Edge-computing-driven internet of things: A survey. ACM Computing Surveys, 2022, 55(8): 1–41. DOI:10.1145/3555308.

[8] Wang R, Lai J, Zhang Z, Li X, Vijayakumar P, Karuppiah M. Privacy-preserving federated learning for internet of medical things under edge computing. IEEE journal of biomedical and health informatics, 2022, 27(2): 854–865. DOI:10.1109/JBHI.2022.3157725.

[9] McEnroe P, Wang S, Liyanage M. A survey on the convergence of edge computing and AI for UAVs: Opportunities and challenges. IEEE Internet of Things Journal, 2022, 9(17): 15435–15459. DOI:10.1109/JIOT.2022.3176400.

[10] Hua H, Li Y, Wang T, Dong N, Li W, Cao J. Edge computing with artificial intelligence: A machine learning perspective. ACM Computing Surveys, 2023, 55(9): 1–35. DOI:10.1145/3555802.

[11] Wen J, Zhang Z, Lan Y, Cu Z, Cai J, Zhang W. A survey on federated learning: challenges and applications. International journal of machine learning and cybernetics, 2023, 14(2): 513–535. DOI:10.1007/s13042-022-01647-y.

[12] Liu Y, Kang Y, Zou T, Pu Y, He Y, Ye X, et al. Vertical federated learning: Concepts, advances, and challenges. IEEE transactions on knowledge and data engineering, 2024, 36(7): 3615–3634. DOI:10.1109/TKDE.2024.3352628.

[13] Zhu J, Cao J, Saxena D, Jiang S, Ferradi H. Blockchain-empowered federated learning: Challenges, solutions, and future directions. ACM Computing Surveys, 2023, 55(11): 1–31. DOI:10.1145/3570953.

[14] Liu J, Huang J, Zhou Y, Li X, Ji S, Xiong H, et al. From distributed machine learning to federated learning: A survey. Knowledge and information systems, 2022, 64(4): 885–917. DOI:10.1007/s10115-022-01664-x.

[15] Ye M, Fang X, Du B, Yuen P C, Tao D. Heterogeneous federated learning: State-of-the-art and research challenges. ACM Computing Surveys, 2023, 56(3): 1–44. DOI:10.1145/3625558.

[16] Srirama S N. A decade of research in fog computing: Relevance, challenges, and future directions. Software: Practice and Experience, 2024, 54(1): 3–23. DOI:10.1002/spe.3243.

[17] Pei J, Liu W, Li J, Wang L, Liu C. A review of federated learning methods in heterogeneous scenarios. IEEE Transactions on Consumer Electronics, 2024, 70(3): 5983–5999. DOI:10.1109/TCE.2024.3385440.

[18] Sharma M, Tomar A, Hazra A. Edge computing for industry 5.0: Fundamental, applications, and research challenges. IEEE Internet of Things Journal, 2024, 11(11): 19070–19093. DOI:10.1109/JIOT.2024.3359297.

[19] Chen J, Yan H, Liu Z, Zhang M, Xiong H, Yu S. When federated learning meets privacy-preserving computation. ACM Computing Surveys, 2024, 56(12): 1–36. DOI:10.1145/3679013.

[20] Yang F, Abedin M Z, Hajek P. An explainable federated learning and blockchain-based secure credit modeling method. European Journal of Operational Research, 2024, 317(2): 449–467. DOI:10.1016/j.ejor.2023.08.040.

Biographies

Zongjian Fu earned a bachelor's degree in communication engineering from Beijing Jiaotong University in 2004. He obtained a master's degree in software technology engineering from University of Electronic Science and Technology in 2010. Currently, he serves as an associate professor at the School of Artificial Intelligence at Zhengzhou Railway Vocational and Technical College. His research areas include computer technology and electronic information technology.