EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Edge Computing Communication Privacy Protection Method Based on Federated Learning Algorithm

Jiamei Xue[1,*] and Wengang Yan[1]

[1]College of Information and Electronic Technology, Jiamusi University, Jiamusi, 154007, China

Abstract

Data heterogeneity, the complexity of privacy budget allocation, and the imbalance between privacy and performance lead to a limited scope of privacy protection constraints and fail to ensure data integrity. Therefore, an edge computing communication privacy protection method based on federated learning algorithm is proposed. Participants use federated learning to locally train the sensing data to obtain a local model, avoiding the interaction of raw data with edge computing nodes and the perception platform. The parameter values of the trained model are perturbed by noise using the adaptive differential privacy technology and uploaded to the edge computing node. The edge computing node performs edge aggregation on the noisy model parameters and uploads them to the perception platform to complete the global aggregation operation, realizing edge computing communication privacy protection. A performance loss constraint mechanism suitable for federated learning is proposed and designed, and the performance loss of the adaptive differential privacy federated model is reduced by optimizing the constraint scope of the loss function, improving the privacy protection effect. Experiments show that this method can effectively add noise to the local model parameters and achieve privacy protection for edge computing communication; the performance loss of this method's privacy protection is small, about 0.4; when transmitting different types of data in edge computing communication, the data integrity after privacy protection processing by this method is above 0.98, with excellent privacy protection effect.

Keywords: federated learning algorithm, edge computing, communication privacy protection, local model, edge aggregation, adaptive differential.

Received on 16 March 2026, accepted on 23 April 2026, published on 04 May 2026

doi: 10.4108/eetsis.12254

1. Introduction

To address issues such as high data transmission latency and large bandwidth consumption, edge computing technology has emerged. This technology deploys computing and data storage on the network edge close to the data source [1], enabling real-time data processing, thus effectively reducing the long-distance transmission of data to the cloud, reducing latency, enhancing real-time response capabilities, and alleviating the computing pressure on the cloud. From in-vehicle computing units in intelligent transportation systems to edge servers in industrial Internet, edge computing has become a key technology supporting the development of emerging fields such as smart cities and intelligent manufacturing [2]. However, while bringing many conveniences, edge computing also raises serious communication privacy protection issues. Since edge computing involves the collection, transmission, and processing of a large amount of users' sensitive data, and these data flow frequently between edge nodes and the cloud [3], the data faces security threats such as being stolen, tampered with, and

*Corresponding author. Email: yanwgQt@163.com

misused during the transmission process. Once users' privacy information is leaked, it will not only cause economic losses and mental distress to individuals, but may also have a serious impact on social order and national security. In the edge computing environment, the data sources are extensive, including various types of sensors, mobile devices, and user terminals, etc. [4], and the formats and types of data are rich and diverse, such as text, images, audio, and video. Different types of data have different characteristics and privacy requirements, and different privacy protection methods need to be adopted. At the same time, the data is dynamic, its generation and transmission are changing in real time, and the data flow and patterns are difficult to predict. This requires that the privacy protection method can adapt to the dynamic changes of the data and adjust the protection strategy in real time to ensure effective protection of data privacy in different scenarios [5].

In recent years, the research on privacy protection for edge computing communication has been increasingly emphasized both at home and abroad. Governments, research institutions, and universities have increased their investment and support in this field. Scholars at home and abroad have made remarkable progress in the theoretical research and technological innovation of edge computing privacy protection. For example, Kaur et al. [6] proposed a privacy protection method based on a dual-channel blockchain, which integrates RSA and AES encryption to ensure the security of data transmission and storage, combines ACL to achieve privacy protection and permission management, uses IPFS off-chain storage to improve scalability, and optimizes data retrieval efficiency through the classification function of smart contracts. However, the optimization of smart contracts is complex and affected by multiple factors. If not optimized properly, it will affect the privacy protection effect. Baawi et al. [7] constructed a lightweight SVM classification model, combined with feature selection, etc. to reduce the exposure of sensitive information, and enhanced privacy protection with differential privacy and secure multi-party computation. However, the model is mainly aimed at malicious code detection, and its applicability is limited. Saravanan et al. [8] used a graph partitioning algorithm to construct a hidden area and proposed the BLH algorithm to achieve efficient construction, enhancing the intensity of location privacy protection. However, little consideration was given to the security of location data transmission and storage. Ranjan et al. [9] proposed a privacy protection method based on authentication and constructed a full-process secure transmission mechanism. However, the mechanism is complex, increasing the implementation difficulty and management cost, and is prone to security vulnerabilities.

The data in the edge computing environment is characterized by diversity and heterogeneity, and the data collected by different edge devices may have different distribution and characteristics. Federated learning allows each edge device to use local data for model training, and then aggregate these model update parameters with different characteristics to obtain a more generalized global model, which greatly improves the privacy protection level of edge computing communication. At the same time, the edge computing environment is dynamic, new data is constantly generated, and the working state of the device may change at any time. Federated learning can support real-time updates of models. When new data is generated by edge devices, the model can be fine tuned locally and the updated parameters can be uploaded to the server to adapt to dynamic environments.

Compared with existing privacy protection methods based on federated learning, the main novelty of this paper lies in: 1) proposing an adaptive differential privacy mechanism that dynamically adjusts the privacy budget based on the historical losses of participants, achieving personalized noise addition instead of using a fixed privacy budget; 2) Designed a performance loss constraint mechanism, derived the loss constraint relationship through Renyi divergence, to minimize model performance degradation while ensuring privacy; 3) We have constructed an edge global two-level perturbation aggregation framework, where edge nodes only handle noisy parameters and the perception platform cannot access the raw data, effectively resisting inference attacks from untrusted servers. These innovations make this method superior to existing methods in terms of privacy protection strength, data integrity, and energy efficiency. In order to verify the progressiveness of this method, compared with recent privacy protection methods based on deep learning, some methods rely on quantum computing resources, and the actual deployment cost is high [10]; Some methods do not consider the semi honest threat of edge nodes [11]. This method does not require additional hardware and aggregates natural defense nodes for eavesdropping through edge perturbation, making it more practical and deployable.

2. Edge Computing Communication Privacy Protection Method

2.1. Edge Computing Communication Privacy Protection Process Based on Federated Learning Algorithm

In the edge computing communication scenario, to avoid the leakage of the original data of participants due to attacks such as untrusted servers, theft, and inference, a privacy protection model for edge computing communication based on the federated learning algorithm is established, as shown in Figure 1.

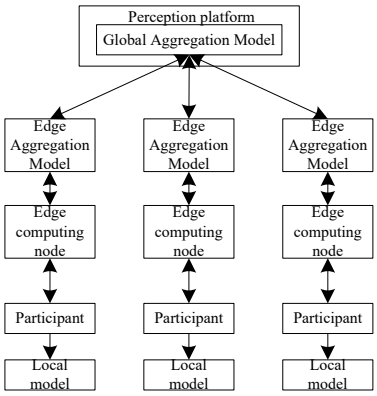


Figure 1. Communication privacy protection model of edge computing based on federated learning algorithm

Using federated learning, an edge computing communication privacy protection model is constructed, which includes four entities: the requester, the participant, the edge computing node, and the sensing platform.

(1) Requester: Initiates an edge computing communication sensing task, interacts with the sensing platform to complete data transactions [12], issues tasks, and obtains the model calculation results after local differential privacy perturbation processing.

(2) Participant: Collects data through edge devices such as intelligent mobile terminals and wearable devices, retains the original data locally to perform model training, randomly perturbs the model parameters using adaptive differential privacy technology, and then transmits the perturbed parameters to the edge computing node through the edge computing communication network.

(3) Edge computing node: As the core of the edge computing communication architecture, it connects participants and the sensing platform, relying on stronger storage and computing resources [13], undertakes edge model training and secure aggregation, and then uploads the edge aggregated model to the sensing platform for further processing.

(4) Sensing platform: As the data processing and control center, it has stronger storage and computing capabilities than edge computing nodes. It collects the noisy edge aggregated models through edge computing communication, conducts global model aggregation and update, and finally returns the results to the requester to complete the task, achieving edge computing communication privacy protection.

The workflow of edge computing communication privacy protection revolves around the interaction among the requester, the participant, the edge computing node, and the sensing platform. The specific steps are as follows:

Step 1: Initialize the sensing task. The requester sets the edge computing communication sensing data collection task and defines the data processing task as a federated learning task [14].

Step 2: Task request. The requester sends an edge computing communication task request to the sensing platform. If there is an initial model, the initial model parameters can be sent in the request [15].

Step 3: Task publication. The sensing platform publishes the federated learning task to the edge computing node [16], and the node screens and invites participants.

Step 4: Local model training and perturbation. The participant uses intelligent devices to collect sensing data, conducts local model training according to the task, and in order to protect the privacy of edge computing communication, adaptively adds noise to perturb the local training model parameters through adaptive differential privacy technology, and then sends them to the edge computing node.

Step 5: Edge perturbation aggregation. The edge computing node receives the noisy local model parameters uploaded by the participant through edge computing communication, performs edge aggregation and update, and then sends them to the sensing platform.

Step 6: Global perturbation aggregation. The sensing platform collects the edge model parameters uploaded by each edge computing node, aggregates and updates them to obtain the global model, and returns it to the edge computing node for the next round of training, or submits it to the requester as the task result.

Step 7: Local model update. The participant obtains the global model of this round from the edge computing node, retrains the local model for the next round, adds local noise perturbation, and iteratively updates until the optimal model is obtained or the maximum number of training times is reached, ensuring the privacy of edge computing communication throughout the process.

2.2. Participant Local Model Training and Perturbation Algorithm for Edge Computing Communication Privacy Protection

In the edge computing communication scenario, participants have a large amount of local data containing personal sensitive information. Through local model training, the data does not need to be uploaded to the sensing platform, reducing the risk of being stolen during transmission and storage [17]. To adapt to the dynamic changes of the environment, the perturbation operation uses adaptive differential privacy technology to add noise to the local model parameters, obscuring the features of the original data. Even if an attacker obtains the noisy model parameters, it is difficult to reverse-engineer the original sensitive data, thus effectively protecting the privacy information of users [18].

In the edge computing communication network, assume there are U participants $\{U_1, U_2, \ldots, U_m\}$, and each participant U_i holds a local dataset $Z_i = \{(x_{i1}, y_{i1}), \ldots, (x_{in}, y_{in})\}$, where x_{ij} is the input

feature, y_{ij} is the label, Z_i is a subset of the global dataset Z, and m is the number of participants. The goal of federated learning is to train the global prediction model $K(w)$, minimize the local loss function $F_i(\omega)$, and anchor the direction for model optimization under privacy protection [19].

The specific steps of local model training are as follows:

Step 1: Local loss function. Participant U_i trains the model based on the local dataset Z_i, and first defines the local loss function as:

$$F_i(\omega) = \frac{\sum_{j \in D_i} f(\omega, Z_{ij})}{|Z_i|} \tag{1}$$

Among them, $f(\omega, Z_{ij})$ is the loss function of the model parameter ω for the j th sample (x_{ij}, y_{ij}). By averaging the losses of all samples in the local dataset Z_i, the local model loss $F_i(\omega)$ of participant U_i is obtained, which is the basis for constructing the subsequent global objective function [20]; D_i is the number of samples.

Step 2: Local model parameter training update [21]. Participant U_i uses the stochastic gradient descent algorithm to update the model parameter ω_i based on the local loss $F_i(\omega)$, realizing the iteration of the local model. The formula is as follows:

$$\omega_i^t = \omega_i^{t-1} - \eta \cdot \nabla F_i(\omega) \tag{2}$$

Among them, ω_i^{t-1} are the model parameters of participant ω_i^{t-1} after the $t-1$ th round of training; η is the learning rate; $\nabla F_i(\omega)$ is the gradient of the local global loss $F_i(\omega)$.

In the edge computing communication privacy protection model described in Subsection 2.1, obfuscating sensitive information through adaptive differential privacy technology is the core step for protecting data privacy in the edge computing communication scenario [22]. The specific steps for perturbing the local model parameters are as follows:

Step 1: Set the initial privacy budget. In the edge computing communication network, there are significant differences in the data quality and privacy requirements of different participants U_i. The sensing platform sets the initial privacy budget $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m\}$ for each participant individually for subsequent addition of Gaussian noise [23]. Among them, ε_i is the initial privacy budget of participant U_i (the smaller the value, the stronger the privacy protection and the greater the noise).

Step 2: Adjust the privacy budget. Before each round of training of participant U_i, the privacy budget is dynamically adjusted according to the loss $F_{i,t}$ in the previous round. Let $F_i^{\max}$ be the maximum historical loss of participant U_i; k is the adjustment coefficient; then the privacy budget adjustment formula is:

$$\varepsilon_{i,t} = \varepsilon_i + k \cdot (F_i^{\max} - F_{i,t}) \tag{3}$$

Among them, $\varepsilon_{i,t}$ is the adjusted privacy budget of participant U_i in the t -th round of training; $F_{i,t}$ is the loss value of participant U_i in the t -th round of training.

Step 3: Add noise perturbation to achieve differential privacy [24]. Gaussian noise is used to add perturbation to the local model gradient to achieve relaxed differential privacy. Let the local model gradient of participant $\nabla F_i(\omega)$ in the t-th round of iteration be $\nabla F_{i,t}$; the sensitivity is $\Delta f = \max_{D_i, D_j} \| f(Z_i) - f(Z_j) \|_2$; among them, the sensitivity Δf of the query function f measures the maximum difference in the query results of adjacent datasets Z_i and Z_j. Let the Gaussian noise distribution be $N(0, \Delta f^2 \cdot \sigma^2)$, σ is the standard deviation; then the noisy gradient is:

$$\nabla \hat{F}_{i,t} \leftarrow \nabla F_{i,t} + N(0, \Delta f^2 \cdot \sigma^2) \tag{4}$$

According to the relaxed differential privacy theorem, if the noise $Y \square N(0, \Delta f^2 \cdot \sigma^2)$ satisfies $\forall \varepsilon \in (0,1), \delta \geq \frac{e^{-\varepsilon\sqrt{2}}}{1.25}$, the condition for the noisy operation to satisfy (ε, δ) differential privacy is:

$$\Pr[M(Z_i) \in Z] \leq e^{\varepsilon} \cdot \Pr[M(Z_j) \in Z] + \delta \tag{5}$$

Among them, M is the differential privacy protection algorithm, and δ is the privacy violation tolerance probability.

Step 4: Transmit and aggregate the perturbed parameters. After participant U_i completes the perturbation, the noisy gradient $\nabla \hat{F}_{i,t}$ is uploaded to the edge computing node through the edge computing communication network, and the node aggregates the noisy gradients of all participants. The formula is as follows:

$$\Delta F_{\text{sum}} \leftarrow \sum_{i=1}^{m} \left[\nabla \hat{F}_{i,t} + N(0, \Delta f^2 \cdot \sigma^2) \right] \tag{6}$$

The sensing platform further aggregates and updates the local model parameters. The formula is as follows:

$$\omega_i^t \leftarrow \omega_i^{t-1} + \frac{\sum_{i=1}^{m}\left[\nabla \hat{F}_{i,t} + N(0, \Delta f^2 \cdot \sigma^2)\right]}{m} \quad (7)$$

Through the above process, based on the local model training, participants complete parameter perturbation relying on the adaptive differential privacy technology, realize the fuzzification of data features, and build a solid privacy defense line for parameter transmission and aggregation in edge computing communication.

2.3. Edge Perturbation Aggregation and Update Algorithm for Edge Computing Communication Privacy Protection

Edge computing nodes in the edge computing communication network are deployed at the network edge, closer to the participants [25], and play a crucial role in the process of computing task execution. Each edge computing node is mainly responsible for collecting the locally perturbed model parameters sent by the corresponding participants in Subsection 2.2 for edge aggregation, and at the same time sending the updated edge model parameters to the sensing platform. All edge computing nodes are semi-honest. They are curious about the private information in the participants' intelligent mobile devices and guess sensitive information [26]. Therefore, the edge computing nodes can only receive the model parameters perturbed by adaptive differential privacy in Subsection 2.2 for aggregation and update.

Before the start of the t -th round of aggregation, the edge computing node downloads the global model parameters ω_{t-1}^{α} of the previous round from the sensing platform, and replaces the locally stored edge model parameters ω_{t-1}^{β} with ω_{t-1}^{α} :

$$\omega_{t-1}^{\beta} \leftarrow \omega_{t-1}^{\alpha} \quad (8)$$

The edge computing node randomly selects γ participants from m participants with probability ρ to participate in this round of aggregation, satisfying $\gamma \leftarrow \max(m\rho, 1)$. The replaced edge model parameters ω_{t-1}^{β} are distributed to the selected participants. Participant U_i performs t_1 rounds of local model updates based on the received ω_{t-1}^{β}, generates a new round of perturbed local model parameters $\tilde{\omega}_i^t$ and local loss function $\tilde{F}_i^t(\omega)$, and sends them back to the edge computing node. The edge computing node aggregates the perturbed local model parameters $\tilde{\omega}_i^t$ of the γ received participants by weighted average according to the dataset size to generate new edge model parameters ω_t^{β} :

$$\omega_t^{\beta} \leftarrow \sum_{i=1}^{\gamma} \frac{|Z_i|}{\hat{N}} \cdot \tilde{\omega}_i^t \quad (9)$$

Among them, $|Z_i|$ is the size of the local dataset Z_i of participant U_i; $|Z|$ is the total amount of training data of the edge computing node.

After the edge computing node executes t_2 rounds of the above aggregation process, it uploads the final edge model parameters ω_t^{β} to the sensing platform, and the aggregated edge loss function $F_t^{\beta}(\omega)$, $F_t^{\beta}(\omega)$ is calculated as follows:

$$F_t^{\beta}(\omega) = \sum_{i=1}^{m}\sum_{i=1}^{\gamma} \frac{|Z_i|}{|Z|} F_{i,t}(\omega) \quad (10)$$

Through the above process, based on local model training and perturbation, the edge computing node realizes the privacy-safe edge model update relying on the weighted aggregation and dynamic participation mechanism.

2.4. Global Perturbation Aggregation and Update Algorithm for Edge Computing Communication Privacy Protection

The sensing platform in the edge computing network, as the data processing center, has stronger storage and computing capabilities than edge computing nodes. It is mainly responsible for receiving the edge model parameters sent by edge computing nodes, aggregating and updating the global model, and providing the final processing results to the requester to complete the sensing task. Since the sensing platform is untrusted, during the global model aggregation and update process, it may use the model parameters to infer the training data of participants through inference attacks, and may also be subject to single-point attacks. Therefore, the sensing platform can only receive the edge-aggregated model parameters with added random noise perturbations sent by edge computing nodes (as in Subsection 2.3), and it has no knowledge of the participants' local model parameters, thus unable to infer the original data of participants and further reducing the risk of sensitive information privacy leakage.

For the model processing process at the sensing platform, at the global loop $t = 0$, the sensing platform initializes the global model parameters ω_0^{α} and sends them to all edge computing nodes, which are broadcast by the edge nodes to participants for initializing local model parameters, providing a unified model starting point for the entire process of federated learning and ensuring the training consistency of each entity. When $t > 0$, every time the sensing platform completes t_2 rounds of edge aggregation, it receives the edge aggregation model

parameters ω_t^β and the edge loss function $F_t^\beta(\omega)$ uploaded by edge computing nodes. The sensing platform weights and averages the aggregation parameters ω_j^β of the j th edge computing node received according to the size of the aggregated dataset of the edge node to generate new global model parameters ω_t^α :

$$\omega_t^\alpha \leftarrow \sum_{j=1}^{J} \frac{\hat{M}_j}{\hat{M}} \cdot \omega_j^\beta \tag{11}$$

Among them, $\hat{M}_j$ is the size of the aggregated dataset of edge computing node β_j; $\hat{M}$ is the total amount of global training data of the sensing platform; J is the number of edge computing nodes participating in training.

The sensing platform synchronously aggregates the loss functions of edge computing nodes to generate the global loss function $F_t^\alpha(\omega)$ for checking the model convergence. The formula is as follows:

$$F_t^\alpha(\omega) = \sum_{j=1}^{J} \frac{\hat{M}_j}{\hat{M}} \cdot F_j^\beta(\omega) \tag{12}$$

If $F_t^\alpha(\omega)$ tends to be stable (e.g., the change in adjacent rounds is less than the threshold), the model is considered to have converged.

The sensing platform distributes ω_t^α to edge computing nodes and continues to execute the next round of edge perturbation aggregation - global perturbation aggregation iteration. When the global iteration round reaches the maximum value (or the model converges), the sensing platform outputs the final global model parameters ω^α to the requester to complete the sensing task (edge computing communication privacy protection).

Through the above process, based on local model training and perturbation as well as edge perturbation aggregation, the perception platform relies on the weighted aggregation and convergence check mechanism to achieve a privacy - secure global model update, and finally outputs a safe and effective perception result to the requester, which is a complete closed - loop of decentralized privacy protection - centralized global optimization in edge - computing communication privacy protection.

2.5. Design of the Performance Loss Constraint Mechanism for Edge Computing Communication Privacy Protection

In the communication privacy protection method of edge computing based on federated learning, although the adaptive differential privacy technology introduced in section 2.2 can improve privacy security, the added noise will interfere with the gradient direction of the model, resulting in slower convergence and lower final accuracy of the model, that is, "performance loss". To quantify and constrain this loss, this section proposes a lightweight performance loss constraint mechanism.

The performance loss caused by privacy protection is essentially due to the deviation of the gradient distribution Q after adding noise from the original gradient distribution P. The larger the difference between the two, the more the model update deviates from the optimal direction, and the greater the performance loss. This article uses Renyi divergence to measure this distribution difference and derives the upper bound constraint relationship of the loss function, so as to dynamically adjust the noise intensity during the training process and achieve a balance between privacy and performance. The specific derivation is as follows:

In the process of edge - computing communication privacy protection, assume there are m participants $U_i, i = 1, 2, \ldots, m$, and r participants are selected from them for federated training at a sampling rate p, and the total number of federated communication iterations is T. After adding noise to the parameters obtained from the local model training of the i - th participant, after T iterations, its privacy loss calculation is related to local training perturbation, edge and global aggregation processes. The calculation formula for the privacy loss (performance loss) of the i - th participant after T iterations is:

$$q_i = \exp(a(t)) = \exp\left(\sum_{t=1}^{T} a(t)\right) \tag{13}$$

Among them, q_i is the performance loss of the i - th participant, reflecting the degree of model performance degradation caused by privacy protection (such as adding noise, etc.); $a(\tau)$ is the performance function of t rounds.

The calculation formula for the overall performance loss Q of the r participants participating in federated training is:

$$Q = \sum_{i=1}^{r} q_i \tag{14}$$

Among them, Q is used to measure the performance loss of the entire training - participating group due to privacy protection.

Let the performance function be written as $a(t) = \log(\max\{E_{v_1,v_0}, E_{v_0,v_1}\})$, which involves two probability density functions. v_0 represents the probability density function of the Gaussian distribution $N(0, \sigma_i^2)$, corresponding to the noise distribution when no additional sampling operations are introduced in local model training and perturbation. v_1 represents the mixed probability density function of $pN(\Delta s, \sigma_i^2) + (1-p)N(0, \sigma_i^2)$,

corresponding to the noise distribution after considering complex scenarios such as participant sampling in edge perturbation aggregation, reflecting the diversity of edge aggregation. The calculation formulas of g_0 and g_1 are as follows:

$$g_0(h) = \frac{1}{\sqrt{2\pi}} e^{-\frac{h^2}{2\sigma_i^2}} \quad (15)$$

$$g_1(h) = \frac{p}{\sqrt{2\pi}} e^{-\frac{(h-\Delta s)^2}{2\sigma_i^2}} + \frac{(1-p)}{\sqrt{2\pi}} e^{-\frac{h^2}{2\sigma_i^2}} \quad (16)$$

Among them, h is a random variable, that is, a continuous variable such as model parameter perturbation, noise or parameter deviation involved in the aggregation process; σ_i^2 is the variance of the noise in the local model training and perturbation of the i th participant; Δs is the offset in the mixed distribution, reflecting the offset of the noise distribution caused by the edge computing communication scenario.

The following definitions are made for E_{v_1,v_0} and E_{v_0,v_1} in the performance function $a(t) = \log(\max\{E_{g_1,g_0}, E_{g_0,g_1}\})$ using the Renyi divergence:

$$E_{g_1,g_0} = E_{h\sim g_1}\left[\left(\frac{g_1(h)}{g_0(h)}\right)^{\lambda}\right] = E_{h\sim g_0}\left[\left(\frac{g_0(h)}{g_1(h)}\right)^{\lambda+1}\right] \quad (17)$$

$$E_{g_0,g_1} = E_{h\sim g_0}\left[\left(\frac{g_0(h)}{g_1(h)}\right)^{\lambda}\right] = E_{h\sim g_1}\left[\left(\frac{g_1(h)}{g_0(h)}\right)^{-\lambda}\right] \quad (18)$$

Among them, $E_{h\sim g}$ represents the expectation of the random variable h following the distribution g, which is used to quantify the expected values of the relevant indicators of the performance loss function under different noise distributions; λ is the order of the Renyi divergence, which is a key parameter controlling the strength of the performance loss constraint.

Through the above complex derivation, the relationship of $E_{g_1,g_0} \geq E_{g_0,g_1}$ can be obtained. According to this relationship, the loss function in the t -th round of the federated training process is further constrained, that is, $a(t) = \log E_{g_1,g_0}$. This formula shows that in the edge computing communication privacy protection method based on federated learning, the loss only needs to be constrained by E_{g_1,g_0}, thereby reducing the performance loss in the federated learning process. While ensuring the privacy security of the local model training, perturbation, edge, and global perturbation aggregation links, the model performance is maintained as much as possible, and the practicality of the overall solution is improved.

2.6. Computational Complexity Analysis

The computational overhead of our method mainly stems from three parts: local model training, adaptive differential privacy noise addition, and edge as well as global aggregation. Let the size of the local dataset of each participant be Di, the dimension of model parameters be d, the number of local iterations per round be E, the number of edge nodes be N, the total number of participants be K, and the number of global rounds be T.

Local training complexity: Each participant performs E stochastic gradient descents per round, with a computational complexity of O(|Di|·d) for each descent. Thus, the complexity for a single participant per round is O(E|Di|d).

Adaptive differential privacy noise addition complexity: Generating a Gaussian noise vector and adding it to the gradient has a complexity of O(d). The privacy budget adjustment formula (3) only involves scalar operations and can be neglected.

Edge aggregation complexity: Each edge node aggregates the noisy parameters from K/N participants under its jurisdiction, with a weighted average complexity of O((K/N)·d).

Global aggregation complexity: The sensing platform aggregates N edge models, with a complexity of O(Nd).

Overall per-round complexity: Since all participants perform computations in parallel, the time complexity per round is max(O(E|Di|d),O(d),O((K/N)d),O(Nd)). As E|Di|≫1 is typically the case, the dominant term is O(E|Di|d), which is of the same order as in standard federated learning. The space complexity is O(d) for storing model parameters. Compared with federated learning without privacy protection, our method only adds O(d) noise addition computation and O((K/N)d) edge aggregation, with the additional overhead being acceptable.

3. Experimental Analysis

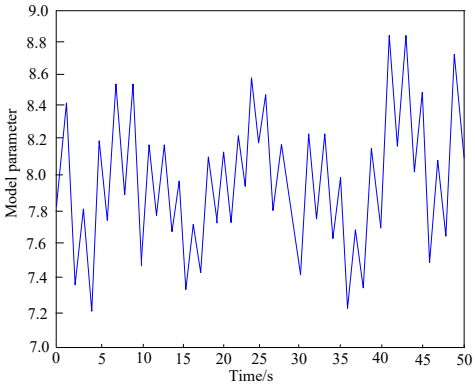
To analyze the effectiveness of the edge computing communication privacy protection method based on the federated learning algorithm in this paper, an edge computing network environment mainly consisting of participants, edge computing nodes, and a sensing platform is constructed based on Python (V3.6.5). Among them, there is 1 sensing platform, 5 edge computing nodes, and 45 participants. It is assumed that each edge computing node has the same number of participants and the same number of training data. In the edge computing network environment deployed in this paper, computing resources are mainly located close to the participants, including terminal devices, so as to better respond to service requests. At the same time, the number of participants can be dynamically changed to ensure the scalability of the edge computing network. During the experiment, PyTorch (V1.0.0) is used to implement the network model and the privacy protection method proposed in this paper. Vectors

conforming to the Laplace distribution are generated by calling the random noise function in Numpy (V1.21.5), and the mean value is set to 0.

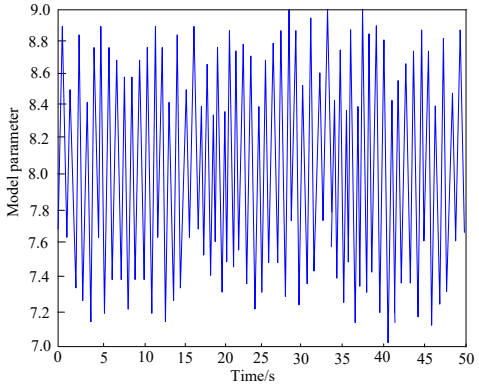
The detailed hyperparameter settings are as follows: Federated learning has 100 global rounds, 5 local iterations per round, a learning rate of 0.01, and a batch size of 64; Adaptive differential privacy: The initial privacy budget is 4.0, the adjustment coefficient is 0.1, the tolerance probability for privacy violations is 10^{-5}, and the Gaussian noise standard deviation is dynamically calculated according to formula (5); Edge aggregation: 30 participants are randomly selected from each round of edge nodes with a probability of 0.8; The optimizer uses SGD with momentum set to 0.9. All experiments were run on servers with Intel Core i9-10900K CPU, NVIDIA RTX 3090 GPU, 64GB RAM, and Ubuntu 20.04 operating system.

The experimental datasets used are the MNIST dataset and the CWRU dataset respectively. Among them, the MNIST dataset contains grayscale image data for 10 kinds of handwritten digit recognition, with 60,000 training images and 10,000 test images. Each grayscale image contains 28 pixels×28 pixels. The CWRU dataset contains vibration signal data of bearings under various operating conditions, covering normal states and different types and sizes of fault states. The normal state records the vibration signals of the bearings in the fault-free state, and different files correspond to different load conditions. The fault types include inner race fault, outer race fault, and rolling element fault. For the outer race fault, it is further divided into 3 o'clock direction (center of the load area), 6 o'clock direction (edge of the load area), and 12 o'clock direction (non-load area). In the experimental analysis, it is assumed that each participant is randomly assigned the same amount of data, and the sensed data samples of the participants are non-independent and identically distributed. Each participant contains two types of samples, and then all participants are randomly assigned to each edge computing node.

The local model parameters of federated learning are processed by adding noise using the method proposed in this paper, and the results of the noise addition are shown in Figure 2.



(a) Local model parameters before adding noise



(b) Local model parameters after adding noise

Figure 2. Local model parameter denoising results

Figure 2 shows the comparison of local model parameters before and after noise addition. Among them, (a) shows regular fluctuations in parameters before adding noise, with parameter values ranging from 7.0 to 9.0; (b) After adding noise, the parameter fluctuations were severe and irregular, and the periodic features were destroyed, verifying the blurring effect of adaptive differential privacy on the original features. By comparing Figure 2 (a) and Figure 2 (b), it can be observed that the parameters after adding noise can no longer directly reflect the precise change trend of the original parameters, which makes it difficult for attackers to infer sensitive information of the original data from the model parameters, thus effectively protecting data privacy. The experimental results show that this method can successfully add noise to the local model parameters, achieving the expected effect of adaptive differential privacy technology. In this way, in the federated learning scenario, each participating party can share model parameters with other participating parties while protecting its own data privacy, jointly train the global model, and promote the secure implementation of federated learning.

Explore the comparison of the performance loss of the method proposed in this paper on the MNIST dataset. Set the number of participating parties to 10, the sampling rate to 1, and the overall privacy budget to 4.0. The analysis results are shown in Figure 3.

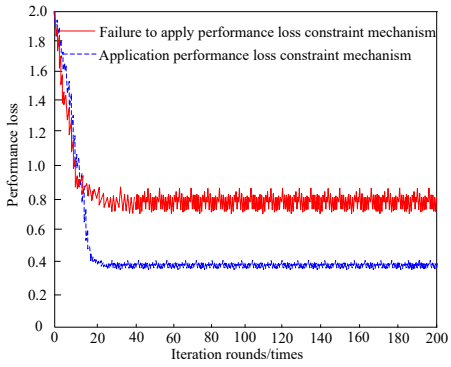


Figure 3. Performance loss analysis results

As can be seen from Figure 3, with the increase of the number of iterative rounds, the performance loss shows a trend of gradually decreasing and tending to be stable. This indicates that during the federated learning process, the model is continuously learning and optimizing, the performance is gradually improving, and the performance loss is gradually decreasing and reaching a relatively stable state. The performance loss curves with and without applying the performance loss constraint mechanism are similar in overall trend, but there are differences in specific values and convergence speeds. In the initial stage (0 - 10 rounds), the performance loss is relatively high, indicating that the initial noise perturbation has a greater impact on model training. The two curves are close, indicating that the constraint mechanism has not yet played a significant role at this time. In the middle stage (10 - 40 rounds), the performance loss drops rapidly, the model gradually adapts to the noise, and the parameter optimization effect is enhanced. The curve with the constraint mechanism drops faster, indicating that this mechanism reduces the ineffective perturbation and improves the training efficiency by dynamically adjusting the noise distribution. In the later stage (>40 rounds), the performance loss tends to be stable, and the model is close to convergence. The final performance loss with the added constraint mechanism is lower, about 0.4, proving that it can better balance privacy and performance in long-term training.

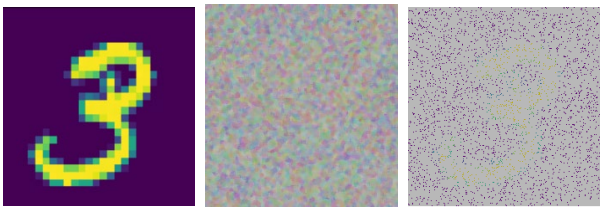
To evaluate the independent contributions of the three core modules of adaptive differential privacy (ADP), performance loss constraint mechanism (PLCM), and edge perturbation aggregation strategy (EAPS), four comparative experiments were designed on the MNIST dataset: 1) complete method (ADP+PLCM+EAPS); 2) Remove PLCM (ADP+EAPS only); 3) Remove ADP (fixed privacy budget only, PLCM+EAPS); 4) Only local training without aggregation (without any privacy protection, as a baseline). Each experiment was repeated 5 times and the average was taken. The overall privacy budget was set to 4.0, with 100 iterations. The final test accuracy, performance loss, and communication energy consumption were recorded. The results are shown in Table 1.

Table 1. Results of ablation experiment

Configuration	Test Accuracy/%	Performance Loss	Communication Energy Consumption/J	Privacy Protection Strength
Complete Method	96.300	0.410	2.3×10^3	High
Without PLCM	94.700	0.580	2.5×10^3	High
Without ADP (Fixed Budget)	95.100	0.520	3.1×10^3	Medium
Without Privacy Protection (Baseline)	97.200	0.000	1.8×10^3	None

According to Table 1, the complete method reduces performance loss by 29.3% and improves accuracy by 1.6% compared to no PLCM configuration, verifying the effectiveness of the performance loss constraint mechanism in controlling the negative impact of noise. Compared with no ADP configuration, the communication energy consumption of the complete method is reduced by 25.8%, because the adaptive privacy budget reduces unnecessary noisy communication overhead, and the privacy protection strength is increased from "medium" to "high". Although the complete method sacrifices about 0.9% accuracy compared to the baseline (no privacy), it results in high-intensity privacy protection (able to resist gradient leak attacks), reflecting an acceptable privacy utility trade-off. In summary, the synergistic effect of the three modules is indispensable.

To verify the privacy protection effect of the proposed method, gradient leakage attacks were used for testing. One image sample and one bearing vibration signal sample were selected from the MNIST dataset and the CWRU dataset for edge computing communication respectively to conduct gradient leakage attacks. The test results are shown in Figures 4 and 5.

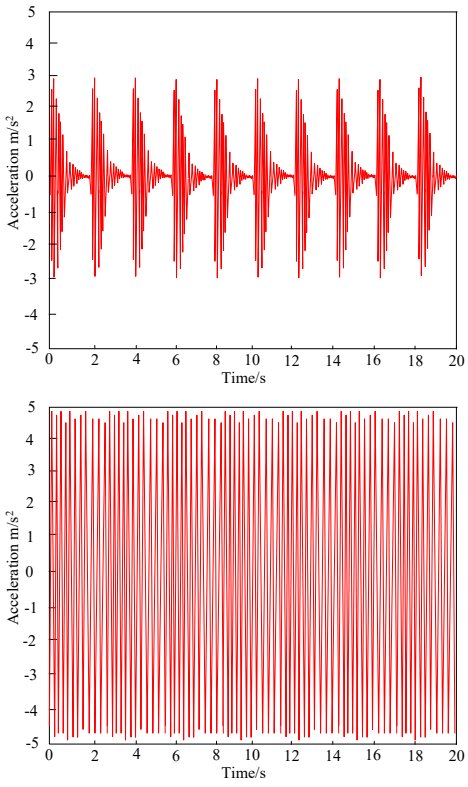


(a) Original image (b) Privacy protection result (c) Gradient leakage attack result

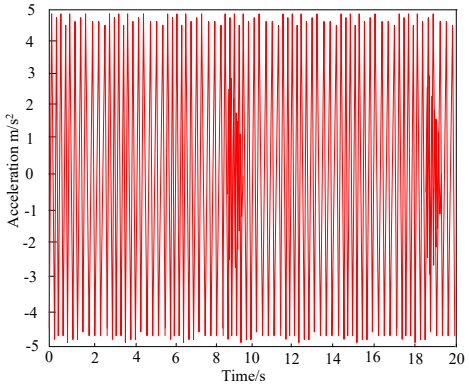
Figure 4. Privacy protection result of edge computing communication image

Analysis of Figures 4(a) to 4(c) reveals that Figure 4(a) displays the original image of the MNIST dataset in edge computing communication, clearly showing the shape of the digit 3. The image has distinct boundaries and continuous pixel distribution, making it easily recognizable by the human visual system. In edge computing communication, such original images, if transmitted without protection, may leak sensitive information. Figure 4(b) shows the image after privacy protection using the proposed method. Compared to the original image, the

protected image exhibits noticeable blurring and noise interference, with the boundaries of the digit 3 becoming unclear and the pixel distribution more dispersed. This indicates that the proposed method effectively processes the original image, altering its original features through privacy protection techniques, making it difficult for attackers to extract useful information directly from the protected image. From a privacy protection perspective, this method reduces the risk of information leakage during image transmission, meeting the basic requirements for data privacy in edge computing communications. Figure 4(c) shows the results after a gradient leakage attack. Compared to the privacy-protected image, the attacked image still partially reveals the shape of the number 3, but its edges are incomplete, and background noise remains. Attackers cannot fully and accurately reconstruct the original image. This indicates that the proposed method can effectively protect image privacy to a certain extent, increasing the difficulty for attackers to obtain real information.



(a) Original bearing vibration signal (b) Vibration signal of bearings after privacy protection



(c) Vibration signal of bearings after gradient leakage attack

Figure 5. Privacy protection result of edge computing communication bearing vibration signal

Figure 5(a) shows the original bearing vibration signal. This signal exhibits typical periodic impact characteristics, with the peak interval approximately around 2s and the acceleration range concentrated around ±3m/s^2, which is in line with the normal operating conditions of the bearing. Figure 5(b) shows the bearing vibration signal after privacy protection processing. Compared with the original signal, the waveform of the protected signal has changed significantly. The original clear periodic characteristics are masked by a large amount of noise interference, the amplitude and shape of the waveform become more irregular, and the acceleration range expands to around ±5m/s^2. This indicates that the proposed privacy protection method effectively processes the original signal. Through the privacy protection method, the statistical characteristics and time-domain characteristics of the signal are changed, making it difficult for attackers to directly extract useful bearing operation information from the protected signal. From the perspective of privacy protection, this method can, to a certain extent, ensure the security of the bearing vibration signal during transmission and reduce the risk of information leakage. Figure 5(c) shows the bearing vibration signal after being attacked by gradient leakage. It can be observed that compared with the signal after privacy protection, some periodic characteristics of the attacked signal are restored, but the peak timing is disordered, and the key fault characteristic frequencies are still covered by noise. This shows that the gradient leakage attack can partially restore the bearing vibration signal after privacy protection. Although the gradient leakage attack can partially restore the signal characteristics, there are still significant differences between the signal after privacy protection and the original signal, and attackers cannot accurately restore the original bearing vibration signal completely. This indicates that the proposed method is effective in protecting the privacy of the bearing vibration signal and can increase the difficulty for attackers to obtain real information. Thus, it can be seen that the proposed

method can effectively resist the gradient leakage attack and protect user data.

Since random noise is imposed on the local model parameters using adaptive differential privacy during the execution of the sensing task for privacy protection, certain losses will be caused to the data information during the training and aggregation of the model parameters, thus affecting the integrity of the sensing data. The smaller the loss, the better the integrity of the data, and the better the availability of the data. In this paper, the adaptive differential privacy technology is used to perform noise perturbation on the local model parameters obtained by participants after each iteration of training. The added random noise may affect the integrity of the data, resulting in the data being unavailable and unable to complete the sensing task. To verify the availability of the data after using the adaptive differential privacy technology for privacy protection in this paper, the influence of the privacy protection degree on the data integrity is studied for different privacy budget values. The analysis results are shown in Figure 6.

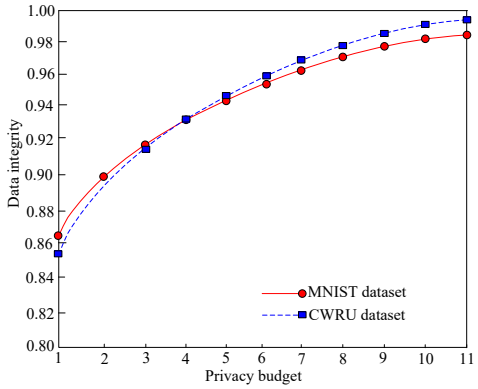


Figure 6. Results of data integrity analysis

As can be seen from Figure 6, with the increase of the privacy budget, the data integrity of the proposed method shows an upward trend. When the privacy budget is between 0 and 5, the slope of the data integrity curve is relatively large. When the privacy budget is between 5 and 11, the curve tends to be flat. This is because the increase of the privacy budget value means that the participants add less noise to the model parameters by differential perturbation locally, thus reducing the difference between the noisy model parameters uploaded by the participants to the edge computing node and the original model parameters. Finally, the data integrity of both datasets is above 0.98, verifying the proposed method's superior ability to balance privacy protection strength and data availability. Its innovative adaptive mechanism provides quantifiable privacy configuration guidance for edge computing scenarios.

The methods in References [6] to [8] and the method in this paper are used for edge computing communication privacy protection processing. The total communication and computing energy consumption of the above five methods during privacy protection processing is analyzed, and the analysis results are shown in Figure 7.

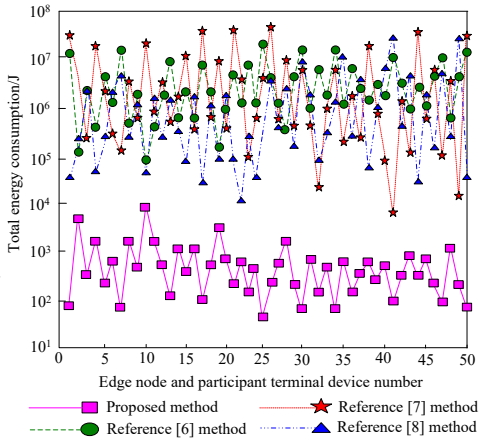


Figure 7. Edge computing communication privacy protection total energy consumption analysis results

As shown in Figure 7, the total energy consumption of the proposed method remains at a relatively low level across all edge nodes and participant terminal device numbers, with the highest total energy consumption being approximately 10^4 J, significantly lower than the methods described in References [6] (10^7 J), [7] (10^8 J), and [8] (10^7 J). The energy consumption fluctuations of the proposed method are relatively stable, with small variations in energy consumption values across different edge nodes and participant terminal device IDs. This reflects the proposed method's good stability and adaptability, enabling it to maintain relatively stable energy consumption performance across different device and node environments. The advantages of the proposed method lie in its use of dynamic participant sampling and gradient compression techniques to reduce communication volume, resulting in a significant decrease in the proportion of communication energy consumption; the application of an edge perturbation aggregation model reduces cloud computing pressure and saves aggregation energy consumption. In contrast, literature [6] experiences an increase in communication energy consumption due to the blockchain consensus mechanism; the lightweight SVM in literature [7] requires frequent model updates, resulting in high computational energy consumption; and while the location hiding algorithm in literature [8] reduces communication load, the complex spatial encryption still incurs high computational overhead. The proposed method can more efficiently balance energy consumption.

In addition, in order to verify the generalization ability of the method on more complex data, the Fashion MNIST data set (including 28 × 28 gray-scale images of 10 types

of fashion items, 60000 training sets and 10000 test sets) and the real edge computing scene are added: based on the open-source EdgeSimPy simulator, the communication environment of 50 roadside units (edge nodes) and 200 networked vehicles (participants) in the intelligent transportation scene is simulated. The data are vehicle tracks and camera images, the transmission bandwidth is set to 20Mbps, and the simulation duration is 30 minutes. Compare the privacy protection effect and communication delay of our method, the method in reference [6], and the baseline (without privacy) in the intelligent transportation scenario simulated by EdgeSimPy. The evaluation indicators include: data integrity (the proportion of available data after privacy protection), average communication delay per round (the time from participant upload to global aggregation completion), and the success rate of resisting inference attacks (the success rate of attackers attempting to restore vehicle trajectories from model parameters). The results are shown in Table 2.

Table 2. Generalization ability on more complex data

Method	Data Integrity	Average Communication Delay (ms)	Success Rate of Trajectory Reconstruction Attack (%)
This method	0.985	127	12.3
Method in Reference [6]	0.912	356	18.7
Without privacy protection	1	89	78.4

According to Table 2, our method maintains a high data integrity of 0.985 in real traffic scenarios, significantly better than the 0.912 in reference [6], because adaptive differential privacy dynamically adjusts noise based on data to avoid excessive disturbance. The communication delay is 127ms, which is higher than the 89ms without privacy protection (due to noisy computation), but much lower than the 356ms in reference [6] (due to high blockchain consensus overhead). The success rate of trajectory restoration attacks is only 12.3%, far lower than the 78.4% without privacy protection, proving that our method can effectively defend against inference attacks targeting moving trajectories. This real-world experiment further validates the practicality and robustness of the proposed method.

In addition to the gradient leakage attack, the defense capability of the method in this paper is evaluated on MNIST and CWRU data sets for the common member inference attack (judging whether a piece of data is in the training set) and model inversion attack (reconstructing training samples from model parameters) in edge computing communication. The member inference attack adopts the shadow model method proposed by Shokri et al., while the model inversion attack adopts the method proposed by Fredrikson et al. The defense success rate is defined as the proportion of samples in which an attack fails (which cannot be accurately inferred or restored). The results are shown in Table 3.

Table 3. Security analysis of different types of attacks

Dataset	Attack Type	Defense Success Rate (%) of Our Method	Defense Success Rate (%) without Privacy Protection	Defense Success Rate (%) of Method in Reference [7]
MNIST	Membership Inference	94.2	31.5	78.3
MNIST	Model Inversion	91.8	15.2	72.6
CWRU	Membership Inference	95.1	28.7	80.1
CWRU	Model Inversion	93.4	18.3	74.5

According to Table 3, the defense success rate of our method under member inference attacks exceeds 94%, which is much higher than that without privacy protection (about 30%) and literature [7] (about 78-80%). This is because the noise added by adaptive differential privacy blurs the output differences of the model on a single sample, making it difficult for attackers to determine whether the sample is participating in training. For model inversion attacks, the defense success rate of our method is about 91-93%, while without privacy protection, attackers can successfully restore about 85% of sample features (i.e., the defense success rate is only 15-18%). The lightweight SVM in reference [7] has a certain anti inversion ability due to its limited model expression ability, but this method further blocks the attack path through edge aggregation and two-level perturbation. In summary, the method proposed in this article can effectively defend against various attacks such as member inference and model inversion, and its security is significantly better than the comparative methods.

4. Conclusion

In edge computing communication scenarios, data privacy protection is crucial. For this reason, we study the communication privacy protection method of edge computing based on the federated learning algorithm. By

training the model locally and adding noise disturbance, we can avoid uploading the original data and effectively reduce the risk of data leakage in the transmission and storage process. At the same time, combined with adaptive differential privacy technology, the privacy budget is dynamically adjusted according to the environment to further enhance the privacy protection effect. The experimental results show that after applying this method, the data integrity is above 0.98, the total energy consumption for privacy protection is less than 10^4 J, and it can resist various attacks such as gradient leakage, member inference, and model inversion.

In practical application, this method can be directly deployed in scenarios such as Internet of Vehicles communication in intelligent transportation system, edge server data aggregation in industrial Internet, and wearable device health data upload in smart medicine. For example, in intelligent transportation, this method can complete the model aggregation of 200 cars within 100 milliseconds while protecting the privacy of vehicle trajectories; In smart healthcare, this method enables hospital edge nodes to collaboratively train disease prediction models without touching patient raw data. This method not only ensures user data privacy, but also promotes the wide application of edge computing communication technology in various fields, and promotes the healthy development of the animal networking and edge computing ecosystem.

Although this method has achieved good results in edge computing communication privacy protection, it still has the following limitations:

1) Limited adaptability to Non IID data: This study assumes that each participant contains two types of samples, belonging to mild Non IID. When the data distribution is extremely skewed (such as each participant containing only one category), the local model bias increases, and the noise adjustment of adaptive differential privacy may fail, resulting in slow convergence of the global model.

2) Relying on a trusted perception platform for global aggregation: Although edge nodes are semi honest, if the perception platform is completely breached, advanced inference attacks may still be launched by analyzing the update patterns of global model parameters. The method proposed in this article assumes that the platform is not trustworthy but will not actively engage in malicious behavior, and has not yet considered the possibility of malicious platforms forging aggregation results.

3) The challenge of computational overhead on resource constrained devices: For extremely low-power edge devices (such as battery powered sensors), performing local model training and denoising operations in each round may result in significant energy consumption. The energy consumption in the experiment is about 2.3×10^3 J, which is still high for some microcontrollers.

Future research directions include: 1) researching federated learning privacy protection methods for extreme Non IID data, introducing clustering or multi center mechanisms; 2) Explore platform verification technologies based on zero knowledge proofs or trusted execution environments to defend against malicious perception platforms; 3) Design lightweight local training algorithms (such as updating only some layers) to reduce device energy consumption; 4) The method in this paper is extended to the layered edge computing architecture (cloud edge end three levels) to further optimize communication efficiency.

Acknowledgements.

This work supported by: Innovation Incentive Program of Science and Technology Plan of Jiamusi:Research on Communication Privacy Protection in Edge Computing Based on Federated Leaming Algorithms (No.GY2025JL0004).

References

[1] Abou El Houda Z, Moudoud H, Brik B, Khoukhi L. Blockchain-enabled federated learning for enhanced collaborative intrusion detection in vehicular edge computing. IEEE Trans Intell Transp Syst. 2024; 25(7):7661-7672.

[2] Zhang L, Wen F, Zhang Q, Gui G, Sari H, Adachi F. Constrained multiobjective decomposition evolutionary algorithm for uav-assisted mobile edge computing networks. IEEE Internet Things J. 2024; 11(22):36673-36687.

[3] Asheralieva A, Niyato D, Miyanaga Y. Efficient dynamic distributed resource slicing in 6g multi-access edge computing networks with online admm and message passing graph neural networks. IEEE Trans Mob Comput. 2024; 23(4):2614-2638.

[4] Jha M K, Kumar M. An autonomic offloading and resource allocation technique for IoT applications in edge computing. J Supercomput. 2025; 81(2):360.

[5] Luo Z, Amayri M, Fan W, Bouguila N. Cross-collection latent beta-liouville allocation model training with privacy protection and applications. Appl Intell. 2023; 53(14):17824-17848.

[6] Kaur J, Rani R, Kalra N. Healthcare data security and privacy protection framework based on dual channel blockchain. Trans Emerg Telecommun Technol. 2025; 36(1):e70049-e70068.

[7] Baawi S S, Oleiwi Z C, Al-Muqarm A M A, Al-Shammary D, Sufi F. Efficient malware detection based on machine learning for enhanced cloud privacy protection. Evol Syst. 2025; 16(1):30.

[8] Saravanan P S, Ramani S, Reddy V R F Y. A novel approach of privacy protection of mobile users while using location-based services applications. Ad Hoc Netw. 2023; 149:103253.

[9] Ranjan A K, Kumar P. APPS: Authentication-enabled privacy protection scheme for secure data transfer in Internet of Things. Ad Hoc Netw. 2024; 164:103631.

[10] Shang W, Ge J, Ding L, Jiang Z, Sui H. Acceleration offloading for differential privacy protection based on federated learning in edge intelligent controllers. Future Gener Comput Syst. 2025; 163:107526.

[11] Qu Z, Zhang L, Tiwari P. Quantum fuzzy federated learning for privacy protection in intelligent information processing. IEEE Trans Fuzzy Syst. 2025; 33(1):278-289.

[12] Herath C, Liu X, Lambotharan S, Rahulamathavan Y. Enhancing federated learning convergence with dynamic

data queue and data-entropy-driven participant selection. IEEE Internet Things J. 2025; 12(6):6646-6658.
[13] Lee S, Zhang T, Prakash S, Niu Y, Avestimehr S. Embracing federated learning: enabling weak client participation via partial model training. IEEE Trans Mob Comput. 2024; 23(12):11133-11143.
[14] Jadav N K, Tanwar S. Whale optimization-orchestrated Federated Learning-based resource allocation scheme for D2D communication. Ad Hoc Netw. 2024; 163:103565.
[15] Ge H, Pokhrel S R, Liu Z, Wang J, Li G. PFL-DKD: Modeling decoupled knowledge fusion with distillation for improving personalized federated learning. Comput Netw. 2024; 254:110758.
[16] Hu X, Cai H, Alazab M, Zhou W, Haghighi M S, Wen S. Federated learning in industrial iot: a privacy-preserving solution that enables sharing of data in hydrocarbon explorations. IEEE Trans Ind Inf. 2024; 20(3 Pt 2):4337-4346.
[17] Al Shahrani A M, Rizwan A, Sánchez-Chero M, Cornejo L L C, Shabaz M. Blockchain-enabled federated learning for prevention of power terminals threats in iot environment using edge zero-trust model. J Supercomput. 2024; 80(6):7849-7875.
[18] Bukhari S M S, Zafar M H, Abou Houran M, Moosavi S K R, Mansoor M, Muaaz M, Sanfilippo F. Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability. Ad Hoc Netw. 2024; 155:103407.
[19] Djukic M, Proki I, Popovic M, Ghilezan S, Popovic M, Proki S. Correct orchestration of federated learning generic algorithms: python translation to csp and verification by pat. Int J Softw Tools Technol Transf. 2025; 27(1):21-34.
[20] Fu T D, Li Y Q. Privacy Protection of Big Data in Complex Network Based on Federated Learning Algorithm. Comput Simul. 2024; 41(6):498-502.
[21] Mafeni V, Kim Y. An automated edge computing approach for iot device registration and application deployment. IEEE Syst J. 2024; 18(2):1447-1458.
[22] Liu F, Liu H, Kannadasan R, Jiang Q. A biometric-based implicit authentication protocol with privacy protection for ubiquitous communication environments. Int J Commun Syst. 2025; 38(1):e5578.
[23] Sarkar S, Agrawal S, Chowdhuri A, Ramani S. Progressive search personalization and privacy protection using federated learning. Expert Syst. 2025; 42(1):e13318.
[24] Zhang J, Si K, Zeng Z, Li T, Ye X. Iea-dp: information entropy-driven adaptive differential privacy protection scheme for social networks. J Supercomput. 2024; 80(14):20256-20582.
[25] Tian X, Du X, Liu X, Wang L, Zhao L. A low-delay source-location-privacy protection scheme with multi-auv collaboration for underwater acoustic sensor networks. IEEE Sens J. 2025; 25(7):12236-12252.
[26] Pei N, Wan B F, Xie S X, Zhang T H, Zhang H F. Yellow light privacy protection with anti-reflection structure based on photonic band gap principle. J Opt. 2024; 26(6):065104.