A Privacy Protection Method for IoT Data Based on Edge Computing and Federated Learning Algorithm

Ying Wu
College of Big Data and Artificial Intelligence,
Zhengzhou University of Economics and Business, Zhengzhou 450000, China

***Abstract*—This paper proposes a privacy protection method for IoT data integrating edge computing and federated learning. To address challenges including edge node heterogeneity, central server bottlenecks in traditional federated learning, and high overhead of homomorphic encryption, we design a hierarchical architecture comprising requesters, participants, edge nodes, a sensing platform, and a key generation center. Participants train models locally using SGD, encrypt parameters with an optimized verifiable dual-key ElGamal homomorphic encryption scheme, and transmit them to edge nodes. Edge nodes employ the MPSDGS algorithm for participant similarity discovery and dropout supplementation, and the MP-Update method for dynamic weighted averaging to ensure continuity and accuracy. Edge-side ciphertext aggregation reduces data volume to the platform. The sensing platform performs global secure aggregation in ciphertext. Experiments demonstrate that the method maintains data privacy above 0.8, with training and aggregation delays within acceptable ranges for typical IoT scales, balancing privacy and efficiency.**

***Keywords*—Edge computing; federated learning; Internet of Things; privacy protection; homomorphic encryption; dropout supplementation**

I. Introduction

The massive devices in the Internet of Things (IoT) generate and transmit sensitive data continuously, providing a rich information foundation for intelligent services [1]. However, the explosive growth of data from these devices includes not only basic data such as device operating status and environmental information but also sensitive information like users' personal privacy and enterprises' commercial secrets. In traditional cloud computing models, data must be centrally uploaded to the cloud for processing and analysis, which not only faces issues such as high latency and bandwidth congestion but also poses significant risks of data privacy leakage [2-3]. Especially in sensitive fields such as healthcare and finance, leaks or misuse of such data could bring severe security risks and economic losses to users [4], and may even trigger social trust crises, hindering the healthy development of the IoT industry [5]. Balancing data utility, service quality, privacy security, and computational efficiency has become a critical challenge in the current IoT field.

To tackle IoT data privacy concerns, scholarly research has introduced diverse conventional approaches. Bezanjani et al. [6] leveraged blockchain-based encryption techniques to secure data transaction workflows while implementing request pattern analysis mechanisms. By integrating multi-source data validation, they identified unauthorized access patterns to proactively mitigate leakage risks. Furthermore, feature optimization combined with Bidirectional LSTM networks enhanced intrusion detection precision against privacy threats. Nevertheless, this approach's excessive dependence on blockchain consensus protocols results in prohibitive computation overhead when handling large-scale IoT data streams, compromising real-time performance requirements. Additionally, deploying sophisticated machine learning models on constrained IoT endpoints creates operational burdens that impede normal device functions and timely data processing. Samriya et al. [7] established multi-layered IoT privacy protection through cloud-level security enhancements using trusted cryptographic analysis approaches. For device-level data processing, they employed structured Markov sparse Bayesian neural networks to extract actionable insights while preserving confidentiality, supplemented by adversarial machine learning for real-time anomaly detection against network intrusions. However, the computational intensity of cryptographic analysis and neural computations creates processing bottlenecks on resource-limited edge devices, necessitating cloud dependency and thereby increasing transmission-related privacy exposure risks. Prakash et al. [8] secured IoT communications through elliptic curve cryptography (ECC) for confidentiality preservation coupled with zero-knowledge proofs (ZKP) for authentication validation. This dual-mechanism approach ensures end-to-end data security by enabling identity verification without content exposure. However, the combined computational complexity of ECC operations and ZKP protocols imposes significant energy consumption and latency penalties on constrained devices, reducing battery efficiency and responsiveness while requiring specialized cryptographic expertise for implementation. Shree et al. [9] integrated blockchain with Inter Planetary File System (IPFS) decentralized storage, employing secret sharing algorithms (SSA) to fragment sensitive data across distributed IPFS nodes. This configuration leverages SSA's information-theoretic security properties to maintain confidentiality even when access keys are compromised, with blockchain integration ensuring transparent data provenance tracking. However, the dynamic accessibility patterns of IoT devices complicate threshold management for data reconstruction from fragmented storage, while the framework provides inadequate protection for edge-side data processing activities.

Edge computing, an emerging paradigm, migrates computing tasks from the cloud to network edge devices,

enabling data processing near data sources [10]. In IoT scenarios, edge computing can preliminarily process and filter locally generated data, uploading only necessary information to the cloud—reducing data transmission and leakage risks [11]. Edge devices with moderate computing power can execute simple encryption and privacy protection algorithms locally, enhancing data security. They also enable local data storage and management, reducing cloud dependency and improving data controllability and privacy. Federated learning, a distributed machine learning framework, allows multiple parties to collaboratively train a global model without sharing raw data. In IoT, devices or edge nodes can act as participants, training models with local data and uploading parameters to a central server for aggregation and global model updates [12]. This avoids centralized storage and transmission of raw data, protecting privacy at the source. Combined with differential privacy and homomorphic encryption, federated learning further strengthens data privacy during model training.

To this end, this paper proposes an IoT data privacy protection method based on the collaborative optimization of edge computing and federated learning. By combining the local processing capability of edge computing with the distributed learning framework of federated learning, a secure and efficient IoT data privacy protection system is constructed. While ensuring the security of data during local processing and transmission, a distributed model training mechanism based on federated learning is developed, allowing each edge node to collaboratively train a global model without sharing raw data, thus protecting data privacy. This provides theoretical support and a technical path for efficient and secure privacy protection in the IoT environment, and promotes the coordinated development of edge intelligence and privacy computing. The main contributions of this paper are summarized as follows: We propose a novel hierarchical privacy protection architecture for IoT data that integrates edge computing, federated learning, and homomorphic encryption, effectively distributing computational loads and mitigating single-point failure risks. We introduce a verifiable dual-key ElGamal homomorphic encryption scheme optimized for resource-constrained IoT devices, employing key segmentation and modular exponentiation optimization to reduce computational overhead. We design the MPSDGS algorithm and MP-Update dynamic weighted averaging method at the edge layer to handle participant dropouts, dynamically allocate aggregation weights based on node capability, and maintain model training continuity and accuracy. We establish a full-process ciphertext operation pipeline from terminal local encryption, through edge ciphertext aggregation, to global ciphertext aggregation on the platform, ensuring data privacy throughout the IoT data lifecycle. Through extensive simulations, we demonstrate that our method achieves a high data privacy degree (above 0.8) while keeping training and aggregation delays within acceptable ranges for typical IoT scales, successfully balancing privacy protection with computational efficiency.

II. Overall Architecture of IoT Data Privacy Protection Based on Edge Computing and Federated Learning Algorithm

To address the challenges of IoT data privacy protection, a privacy protection system model integrating edge computing, federated learning, and homomorphic encryption technology is constructed. This model ensures privacy security throughout the entire process of IoT data sensing, transmission, and processing, achieving a balance between efficient computing and privacy protection [13]. To prevent leakage of participants' original sensing data during sensing tasks, homomorphic encryption privacy protection technology is integrated into the distributed edge computing network. The overall architecture of IoT data privacy protection based on edge computing and federated learning algorithm, as shown in Fig. 1, includes five core entities:

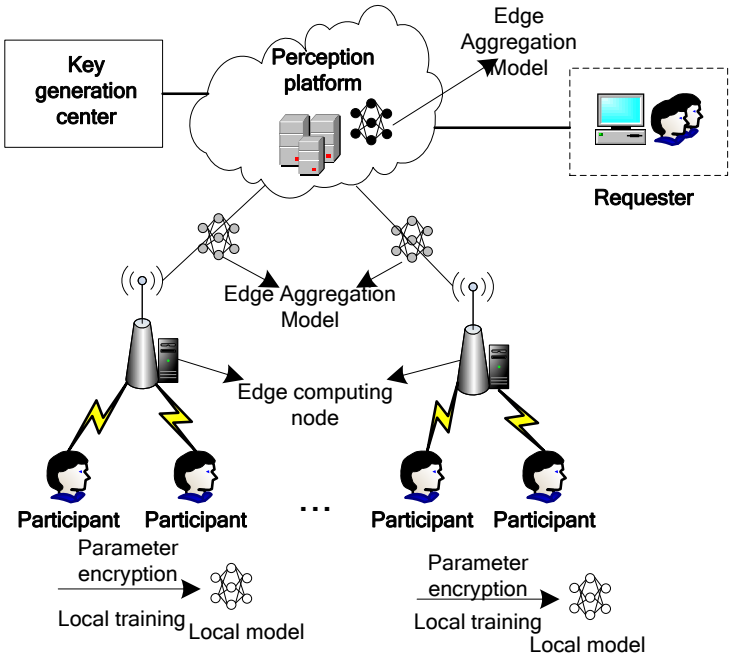

Fig. 1. Overall architecture of IoT data privacy protection based on edge computing and federated learning algorithm.

1) Requester: As the initiator of sensing tasks and consumer of sensing data, it issues task requests to the sensing platform, executes sensing tasks, obtains encrypted training model results, and completes data applications in the IoT.

2) Participant: As the producer of sensing data, it collects sensing data using intelligent mobile devices, performs local federated learning model training, encrypts model parameters using verifiable dual-key ElGamal homomorphic encryption technology, and transmits the encrypted parameters to edge computing nodes via wireless networks.

3) Edge computing node: As an intermediate processing unit for sensing tasks, it has more sufficient storage and computing resources than participants' mobile devices and is deployed at the network edge. It receives encrypted model parameters from participants, executes the MPSDGS algorithm and MP-Update method for similarity calculation and dropout supplementation, performs edge aggregation, and then sends the updated edge model parameters to the sensing platform.

4) Sensing platform: As the data processing and control center, it has powerful storage and computing capabilities. After receiving encrypted edge model parameters from edge computing nodes, it uses homomorphic encryption technology to complete global model aggregation and updates in ciphertext form [14], and finally feeds back results to the requester in encrypted form to complete the sensing task.

5) Key generation center: As a trusted authority, it is responsible for generating and distributing certificate-equipped keys adapted to verifiable dual-key ElGamal encryption to all entities. It provides support for secure training and aggregation processes, ensuring that participants' privacy is not leaked during task collaboration.

B. Local Training and Homomorphic Encryption of Parameters

In distributed machine learning scenarios, participants first conduct local model training using the Stochastic Gradient Descent (SGD) algorithm. Due to its characteristic of updating parameters sample by sample, SGD can effectively handle large-scale data and improve training efficiency. After completing local training, to ensure the security and privacy of model parameters, the verifiable dual-key ElGamal homomorphic encryption technology is adopted to encrypt the model parameters. This encryption method not only has homomorphic properties, allowing specific operations to be performed on ciphertexts, but also can ensure the correctness of the encryption process through a verifiable mechanism. Considering the limited computing power of edge devices, the encryption process is optimized to adapt to their computing capabilities. On one hand, key segmentation technology is used to split the key into multiple parts, reducing the complexity of managing a single key and the computational pressure. On the other hand, through the optimization of modular exponentiation operations, the number of modular operations in the computing process is reduced, thereby effectively lowering the complexity of encryption computation. Finally, the encrypted model parameters are transmitted to edge nodes, enabling secure and efficient model aggregation and updates.

1) Local model training: Traditional distributed machine learning requires concentrating the local data of participating nodes in a server for training. However, the federated learning algorithm applied in this paper keeps the data local to avoid privacy leakage. Participants use the Stochastic Gradient Descent (SGD) algorithm for local training to minimize the loss function. The SGD algorithm is chosen for its efficiency in handling large-scale datasets and its suitability for the iterative, distributed nature of federated learning. Its sample-by-sample update characteristic makes it computationally feasible for resource-constrained IoT devices. Define the N participants in federated learning as $P=\{P_1,P_2,\cdots,P_N\}$. The local data labels of a single participant P_k as $d_k=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$, where x_i is the input parameter and y_i is the expected output. The label dataset of all participants is $D=\bigcup d_i$. Let the model parameters of P_k trained locally be $\omega_k=\{\omega_{k1},\omega_{k2},\cdots,\omega_{km}\}$. The goal of federated learning is to obtain the globally trained model $M_G=h_\omega(x^i)$, so as to minimize the loss function $L(M_G)$ of the dataset D [15].

The loss function of a single participant P_k for the data label d_k is defined as:

$$L(h_{\omega_k}(x^i))=\frac{1}{|d_k|}\sum_{j\in d_k} f_j(h_{\omega_k}(x^i),y_j) \tag{1}$$

Among them, $f_j(h_{\omega_k}(x^i),y_j)$ is the loss function of the data label (x_j,y_j) based on the model $h_{\omega_k}(x^i)$. In T iterations, the training goal of participant P_k is to optimize the local model $h^*_{\omega_k}(x^i)$ under the condition of privacy protection to minimize the loss function, that is:

$$\begin{aligned} &h^*_{\omega_k}(x^i)=\underset{\omega_k\in\{\omega_k(t)\}_{t\le T}}{\arg\min}\ L(h_{\omega_k}(x^i))\\ &s.t.\Pr(\omega_k\in R_d)\le e^{\int}\Pr(\omega_{k'}\in R_d)\\ &\forall P_k\in P,k\in(1,2,\cdots,N) \end{aligned} \tag{2}$$

Among them, $\omega_k(t)$ is the parameter set for the t-th round of joint training, and T is the maximum number of parameter update iterations; $\Pr(\omega_k\in R_d)\le e^{\int}\Pr(\omega_{k'}\in R_d)$ is the differential privacy condition for updating the parameter ω_k.

Participants use the Stochastic Gradient Descent (SGD) algorithm for local training to minimize the loss function. The gradient calculation formula is:

$$\nabla L(h_{\omega_k}(x^i))=\frac{\partial L(h_{\omega_k}(x^i))}{\partial\omega_k} \tag{3}$$

In the t-th iteration of participant P_k, the update of model parameters is defined as:

$$\omega_k(t) = \omega_k(t-1) + \alpha_t \cdot \nabla L(h_{\omega_{k_t}}(x^i)) \quad (4)$$

Among them, α_t is the learning rate, moving in the direction opposite to the gradient of the loss function to approach the optimal result.

In addition, during local training of participants, local data is trained according to the global model. Each participant, based on the initial global model parameter ω_t and local data, uses small-batch stochastic gradient descent for optimization. Through forward propagation, loss calculation, back propagation, and parameter update, a new local model ω_{t+1}^k is obtained. The process is as follows:

$$\begin{cases} \omega_{t,0}^k = \omega_t \\ \omega_{t,\tau+1}^k = \omega_{t,\tau}^k - \eta * g_{t,\tau}^k, (\forall \tau = 1, \cdots, E) \\ \omega_{t+1}^k = \omega_{t,E}^k \end{cases} \quad (5)$$

Among them, η is the learning rate, $g_{t,\tau}^k$ is the random gradient of the small-batch data sample, and E is the number of local training epochs.

2) Parameter homomorphic encryption: Federated learning provides basic privacy protection for raw data due to its local training nature. However, participants may still suffer malicious attacks leading to information leakage, so encryption mechanisms are needed for enhanced protection. A verifiable dual-key ElGamal homomorphic encryption method, which combines Feldman's Verifiable Secret Sharing (VSS) and ElGamal encryption, is adopted to encrypt and protect the model parameter ω_{t+1}^k obtained from participants' local training. The specific process is as follows:

a) System initialization and key generation: The Key Generation Center (KGC), as a trusted third-party, performs the following operations:

- Select large prime numbers p and q satisfying $q | p-1$, and choose a generator g in the multiplicative group L_p^* modulo p;
- Generate a public-private key pair (sk_k, pk_k) for each participant P_k, where $sk_k \in_R L_q$ and $pk_k = g^{sk_k} \bmod p$;
- Generate the global public key $PK = \prod_{k=1}^{N} pk_k \bmod p$, and make (p, q, g, PK) public.

b) Parameter encryption process: After participant P_k completes local model training to obtain the parameter ω_{t+1}^k, the following encryption steps are executed:

- Convert the model parameter ω_{t+1}^k into an integer vector $\omega_{t+1}^k = [\omega_{t+1,1}^k, \omega_{t+1,2}^k, \cdots, \omega_{t+1,m}^k]$;
- For each parameter component $\omega_{t+1,j}^k$, select a random number $r_j \in_R L_q$;
- Calculate the encrypted parameter $Enc(\omega_{t+1,j}^k) = (c_{1,j}, c_{2,j})$, where $c_{1,j} = g^{r_j} \bmod p$ and $c_{2,j} = \omega_{t+1,j}^k \cdot PK^{r_j} \bmod p$;
- Send the encrypted parameter vector $Enc(\omega_{t+1}^k) = [Enc(\omega_{t+1,1}^k), Enc(\omega_{t+1,2}^k), \cdots, Enc(\omega_{t+1,m}^k)]$ to the edge computing node.

Through the above-mentioned encryption process, the local model parameter ω_{t+1}^k of participant P_k is converted into the ciphertext form $Enc(\omega_{t+1}^k)$, which has semantic security, so that an attacker cannot infer the original parameter value from the ciphertext.

C. Edge-Side Secure Aggregation Model

In the edge computing scenario, to enhance the efficiency and security of model aggregation, edge computing nodes adopt a series of innovative strategies. First, the MPSDGS algorithm, which is based on participant similarity discovery and a dropout-supplementation mechanism, is incorporated. This algorithm can accurately identify similar participants and construct a better aggregation group. Meanwhile, when a participant drops out, its dropout-supplementation mechanism can quickly find a suitable replacement, ensuring the continuity of the aggregation process. Then, the MP-Update dynamic weighted-average method based on recursive update rules is used to supplement dropped-out participants, and aggregation weights are dynamically allocated according to the computing power of edge nodes. This measure avoids low-computing-power nodes from becoming performance bottlenecks due to insufficient processing capacity, guarantees the high efficiency of overall aggregation, and maintains a relatively high aggregation accuracy [16]. In terms of privacy protection, the integrity of privacy protection is enhanced, and all operations are carried out under the premise of strictly ensuring data privacy. After that, edge-side model aggregation is completed in the ciphertext form. The edge computing node, as a local aggregation center, performs preliminary aggregation on participants' model parameters. This process can not only effectively reduce the amount of data uploaded to the sensing platform, lowering the security risks during data transmission, but also significantly reduce the computing pressure on the center. As a result, the entire model training and aggregation process becomes more efficient and stable, meeting the strict requirements for real-time performance and security in the edge computing environment.

Privacy Protection Optimization Based on Participant Dropout Resolution Mechanism

The MPSDGS algorithm is described in the following pseudocode (Algorithm 1):

Algorithm 1: MPSDGS (Participant Similarity Discovery and Dropout Supplementation)

Input: Participant status list StSt, Local model parameters θitθit for online participants PiPi
Output: Supplemented participant set for aggregation
1: Initialize similarity list Sim←[]Sim←[], corrected similarity list CorrSim←[]CorrSim←[]
2: for each online participant PiPi do
3: for each online participant PjPj (j ≠ i) do
4: Calculate Pearson correlation coefficient ρijρij using Eq. (6)
5: Sim.append(ρij)Sim.append(ρij)
6: end for
7: end for
8: for each participant PkPk do
9: CorrSim[k]←St[k]×Sim[k]CorrSim[k]←St[k]×Sim[k] // Element-wise multiplication with status list
10: end for
11: for each offline participant PoffPoff do
12: Find online participant PonPon with max CorrSimCorrSim relative to PoffPoff
13: Supplement parameters: θofft←θontθofft←θont
14: end for
15: return Supplemented participant parameter set

1) Privacy protection optimization based on participant dropout resolution mechanism: In the IoT environment, federated learning relies on the collaborative training of local data from multiple participants. However, participants are prone to dropping out due to network fluctuations, device failures, and other factors, resulting in data loss and abnormal model convergence, which weakens the privacy protection capability. To address this, edge computing nodes adopt the MPSDGS algorithm (based on participant similarity discovery and dropout supplementation) combined with the MP-Update dynamic weighted-average method. From the two aspects of data integrity restoration and accurate similarity evaluation, the continuity of model training is guaranteed, laying a solid foundation for privacy protection.

a) MPSDGS algorithm and participant similarity calculation: Edge nodes generate a status list $S=[s_1,s_2,\cdots,s_K]$ (where K is the total number of participants) based on the upload status and validity verification results of participants' encrypted parameters, marking online/offline statuses. When there are offline participants, the MPSDGS algorithm uses the Pearson correlation coefficient to measure the similarity of local model parameters among participants [17], providing a basis for offline participant supplementation. For online participants i、j, edge computing nodes calculate the linear correlation degree Δ_t^i, Δ_t^j based on the uploaded local model parameters $p(\Delta_t^i,\Delta_t^j)$. The formula is:

$$p(\Delta_t^i,\Delta_t^j)=\frac{\sum_{i=1,j=1}^{n}\left[(\Delta_t^i-\mu_i)-(\Delta_t^j-\mu_j)\right]}{\sqrt{\sum_{i=1}^{n}(\Delta_t^i-\mu_i)^2}\sqrt{\sum_{j=1}^{n}(\Delta_t^j-\mu_j)^2}} \tag{6}$$

Among them, μ_i,μ_j are the sample means.

Offline supplementation maintains training continuity through three steps, indirectly safeguarding IoT data privacy:

- Satus Recognition: Edge computing nodes generate a participant status list $[1,0,0,1,1...]$ (1 for online, 0 for offline) based on whether encrypted model parameters are uploaded. The encrypted parameter transmission link is protected by technologies such as homomorphic encryption, and the recognition process does not disclose raw data.
- Similarity Correction: Multiply the status list and the Pearson similarity list element-by-element to set the similarity of offline participants to 0, avoiding interference from invalid similarity, ensuring the accuracy of supplementary data, and indirectly reducing the risk of privacy leakage caused by the introduction of wrong parameters.
- Parameter Supplementation: Screen the online participant corresponding to the maximum corrected similarity and replace the offline participant with its model parameters. The supplementation process operates based on encrypted parameters (such as the edge node aggregation process), ensuring the privacy of data transmission and use, achieving the effect of full participation of all participants, and maintaining the integrity of model training.

b) MP-Update dynamic weighted averaging: Due to the randomness of the initial model and the uncertainty of small-batch gradient descent, the Pearson similarity in a single round cannot accurately reflect participant associations. The MP-Update method is based on recursive update rules [18], dynamically adjusting weights by combining historical and current similarities. The formula is:

$$P_t^{i,j}=\begin{cases}\frac{1}{2}P_{t-1}^{i,j}+\frac{L_{t-1}^{i,j}}{2L_{t-1}^{i,j}+1}P_{t-1}^{i,j}+\frac{1}{2L_{t-1}^{i,j}+1}p_t^{i,j} & i,j\in M_t\\ \frac{1}{2}P_t^{i,j}+\frac{1}{2}p_{t-1}^{i,j} & otherwise\end{cases} \tag{7}$$

Among them, $L_{t-1}^{i,j}$ is the number of rounds where both participants were online before round $t-1$, and $P_{t-1}^{i,j}$ is the average similarity of the previous round.

This method dynamically balances historical data and current states to accurately evaluate participant similarity without exposing raw data, enabling the server to identify the most suitable online participants for dropout replacement. This approach enhances model convergence efficiency while avoiding privacy leaks caused by incorrect similarity matching,

deeply integrating with privacy-protection components such as encryption and aggregation.

In summary, the participant dropout resolution mechanism employed by edge computing nodes supports the privacy-protection system of IoT federated learning by ensuring training continuity and optimizing similarity evaluation. It is an indispensable robustness-enhancement module in the privacy-protection framework.

2) Encrypted parameter aggregation and verification: After completing dropout replacement, edge computing nodes collect the encrypted local model parameters from participants, perform aggregation and verification in ciphertext form, and send the updated edge model parameters to the sensing platform [19] without accessing the actual local model parameters, thereby protecting participant privacy.

Under the edge computing network, the process of aggregating and verifying the encrypted parameters of edge computing nodes is as follows (for the t th training round):

- The edge computing node downloads the encrypted global model parameters $Enc(\omega_{t-1}^{(global)})$ from the previous round from the sensing platform, replaces its existing model parameters $Enc(\omega_{t-1}^{(edge)})$, and sends them to the selected participants.
- Randomly select $k \leftarrow \max(K, \rho 1)$ participants at a proportion of ρ to form a set S_t, and send $Enc(\omega_{t-1}^{(global)})$ to these participants.
- After the participants complete local encrypted training in round t_1, the edge computing node collects the new local model parameters $Enc(\omega^i)$ and the local loss function $F_i^{(t)}(\omega)$.
- Since the decryption key cannot be obtained, only algebraic operations are performed on the ciphertext. The received $Enc(\omega^i)$ is weighted and averaged to achieve edge-side secure aggregation and update:

$$Enc(\omega_t^{(edge)}) \leftarrow \sum_{i \in S_t} \frac{N_i}{N} Enc(\omega^i) \tag{8}$$

Among them, N_i is the size of the dataset D_i of the i-th participant, and $N = \sum_{i=1}^{k} N_i$, which reflects the additive homomorphic property of homomorphic encryption.

When the number of local updates t_1 meets the conditions, the edge computing node sends a signal to the participants to stop training, and sends the encrypted edge model parameters $Enc(\omega_t^{(edge)})$ and the loss function $F_t^{(edge)}(\omega)$ to the sensing platform; if the conditions are not met, the aggregated edge model parameters are sent to the newly selected participants, and the local model update and encryption continue.

D. Global Secure Aggregation Model

The sensing platform, as the main control center for sensing tasks, provides resources for complex data processing and long-term storage. It does not directly interact with participants. It receives the encrypted model parameters aggregated and updated by edge computing nodes, follows the cryptosystem of verifiable dual-key ElGamal encryption [20], and performs global aggregation and update in ciphertext form. This addresses the issue of sensitive data leakage caused by single-point failure attacks and inference attacks, and trains an ideal application model for sensing tasks.

The algorithm flow for global secure aggregation and update is as follows:

- When the global cycle $t = 0$, the sensing platform receives the task, initializes the model parameter ω_0 and sends it to all edge computing nodes, which then distribute it to participants to initialize local model parameters and conduct preliminary training.
- When $t > 0$, the sensing platform sends the latest encrypted global model parameters to edge computing nodes for collaborative training with participants.
- After every t_2 rounds of edge secure aggregation updates, the sensing platform receives the encrypted edge model parameter results $Enc(\omega_t^{(edge)})$ and performs global aggregation and averaging in ciphertext form:

$$Enc(\omega_t^{(global)}) \leftarrow \sum_{j=1}^{J} \frac{M_j}{M} Enc(\omega_t^{(edge)}) \tag{9}$$

Among them, J is the number of edge computing nodes participating in training, M_j is the size of the aggregated dataset of the j-th edge computing node, $M_j = N$, and $M = \sum_{j=1}^{J} M_j$, reflecting the additive homomorphic property of homomorphic encryption.

The sensing platform receives the edge loss function $F_t^{(edge)}(\omega)$, aggregates and averages it to obtain the global loss function $F_t^{(global)}(\omega) = \sum_{j=1}^{J} \frac{M_j}{M} F_t^{(edge)}(\omega)$. It then sends the latest encrypted global model parameters $Enc(\omega_t^{(global)})$ to edge computing nodes for iterative training [21]. When the global loss function converges or the maximum number of model update rounds is reached, the sensing platform sends a signal to edge computing nodes to stop training, and sends the encrypted global model parameters to requesters to complete the sensing task.

In summary, the global secure aggregation model achieves a privacy protection loop among the sensing platform, edge nodes, and terminal participants by combining ciphertext-state collaborative training and hierarchical key management. On one hand, relying on the additive homomorphic property of homomorphic encryption, it ensures that model parameters are

usable but invisible during the global aggregation stage, avoiding platform-side model reverse-attack risks from the system-top level. On the other hand, through multi-round iterative ciphertext aggregation and loss-function convergence control, it ensures that IoT data remains encrypted throughout the entire IoT data flow process. This builds a multi-level, full-link privacy protection system for terminal device sensitive data, edge-node aggregated features, and platform global model parameters.

III. Experimental Analysis

To comprehensively evaluate the application effect of the proposed IoT data privacy protection method, simulations are conducted focusing on two core aspects: privacy protection capability and computational efficiency (delay). While the primary goal is robust privacy preservation, its practical deployment in resource-constrained IoT environments necessitates an assessment of the associated computational overhead and time delays. Therefore, the experiments are designed to verify that the proposed method achieves a favorable balance between security and efficiency, ensuring that strong privacy protection does not come at the cost of unacceptable performance degradation for typical IoT services.

To verify the application effect of the IoT data privacy protection method based on edge computing and federated learning algorithm proposed in this paper, a privacy protection system in the edge computing network environment is simulated. A three-layer IoT network architecture of user layer-edge layer-perception layer is built using Python to simulate the proposed system. One key generation center is set up to generate and manage public and private keys,providing cryptographic support for the training and encryption processes of federated learning in the edge computing network. One sensing platform (simulating the IoT application server) is configured to be responsible for global model aggregation and task scheduling, connecting 6 edge computing nodes (simulating edge computing power carriers). Each node is connected to 14 IoT terminal participants (such as smart sensors, health monitoring devices, etc.), with a total of 84 participants, supporting a dynamic dropout rate of 10%-20% to simulate scenarios such as network fluctuations and failures of IoT devices. The participant data is set with non-independent and identically distributed (Non-IID) settings, and each participant is only assigned 2 types of local data samples to reproduce the locality and heterogeneity of IoT terminal data collection. The federated learning model and the verifiable dual-key ElGamal homomorphic encryption algorithm designed in this paper are implemented using PyTorch (V1.13.1). The private key is only distributed to participants, ensuring that participants can encrypt model parameters locally, resisting the semi-honest attack risks of edge computing nodes and the sensing platform, which conforms to the privacy protection requirements of data cross-level transmission in the IoT environment.

The experimental dataset uses a custom industrial sensor dataset to simulate terminal-collected data. This dataset is collected by IoT sensors (acceleration sensors, temperature and humidity sensors, electromagnetic sensors, etc.) deployed in industrial sites, including 100,000 training samples and 15,000 test samples. Each sample is a multi-dimensional feature vector: covering the vibration frequency, temperature and humidity, valve switch state, etc. during equipment operation, and the label is the equipment health state (0 for normal, 1 for abnormal warning), used for the abnormal detection classification task.

The hardware environment of the sensing platform and edge nodes is: Intel (R) Xeon (R) CPU E5-2630 v3 @ 2.50GHz, 128GB RAM (without GPU), matching the computing power of the edge and the platform; the hardware environment of the participant terminal is: Raspberry Pi 4B (quad-core Cortex-A72 @ 1.5GHz, 4GB RAM), simulating the resource-constrained characteristics of IoT terminals; the operating system is Ubuntu 20.04, ensuring cross-device compatibility. The key parameter settings are shown in Table I, covering the training and encryption of federated learning and the adaptation requirements of the IoT scenario.

TABLE I. Details of Key Parameter Settings

Parameter	Numerical Value
Global aggregation times	30
Edge aggregation times	5
Local iteration times	20
Batch size of data	32
Learning rate	0.01
Learning rate decay rate	0.99
Learning rate decay rounds	3
SGD momentum	0.9
ElGamal key length	2048 bits
Pearson similarity calculation window	5 rounds

TABLE II. Simulation Details of Multi Level Attacks in the Internet of Things

Attack Level	Attack Type	Attack Target (privacy theft content)	Simulate Attack Methods
Terminal side	Local data theft attack	Raw sensitive data collected by the terminal (abnormal values of equipment vibration)	Implant memory reading scripts into Raspberry Pi terminals, attempting to extract local model parameters and raw data cache before encryption
Edge side	Cipher reasoning attack	Distribution of abnormal states of terminal devices (proportion of abnormal production line vibrations)	Deploy ciphertext analysis tools at edge nodes, input edge aggregated ciphertext, use clustering algorithms and homomorphic encryption to crack scripts, and infer terminal data features in reverse
Platform side	Model reverse attack	Full terminal privacy distribution (device health profile)	Run the model reverse engineering algorithm on the platform server, input global model parameters+IoT terminal topology information, and infer privacy profiles

To verify the privacy protection ability of the method proposed in this paper in the IoT data privacy protection under the edge computing and federated learning algorithm, the full process of terminal local training → edge aggregation → platform global training is completed according to the experimental settings, and the intermediate data of each level (local parameters before terminal encryption, ciphertext parameters after edge aggregation, platform global model) are saved. According to Table II, simulate semi-honest attacks and malicious inference attacks at each level of IoT terminal → edge → platform, and verify whether the method can protect privacy information such as device status and sensitive features in the actual data flow, and effectively resist multi-level threats of the IoT. The privacy protection results are shown in Table III.

TABLE III. PRIVACY PROTECTION RESULTS OF THE METHODS IN THIS PAPER

Attack Level	Protection Effect	Core Protection Process
Terminal side	The attacker implanted a memory read script and was unable to obtain valid sensitive information about abnormal device vibration values.	The private key is only directed to the terminal, and the local model parameters of the terminal are encrypted with a verifiable dual key ElGamal before uploading, blocking the exposure of sensitive information from the source of the data
Edge side	The deployment of ciphertext analysis tools on edge nodes cannot effectively infer the distribution of abnormal states of terminal devices after inputting aggregated ciphertext.	The edge aggregation process is based on homomorphic encryption characteristics to perform ciphertext operations, combined with a verifiable dual key mechanism, to resist inference attacks such as ciphertext parameter clustering analysis and cracking scripts
Platform side	The platform server running model reverse engineering algorithm, combined with the topology information of IoT terminals, cannot restore the true privacy distribution of all terminals.	The global model aggregation stage maintains the ciphertext form, relying on the hierarchical key management mechanism of the key generation center to block the expansion of global model parameters and terminal deployment locations

By simulating typical attack scenarios on the IoT terminal side, edge side, and platform side in Table II, the privacy protection results of the method in Table III verify the effectiveness of the method in multi-level privacy protection. On the terminal side, through the localized deployment of verifiable dual-key ElGamal encryption, the original sensitive data is encrypted and uploaded, building a privacy protection barrier from the source of data collection, and greatly reducing the risk of privacy leakage caused by the intrusion of terminal devices; on the edge side, using the ciphertext operation characteristics of homomorphic encryption and the dual-key verification logic, it resists ciphertext inference attacks in the edge aggregation link, ensuring that privacy information such as the abnormal state distribution of terminal devices is not reversely deduced; on the platform side, through ciphertext-state global aggregation and hierarchical key management, it blocks the associated attack path between model parameters and IoT topology information, preventing the reverse engineering of the full-volume terminal privacy distribution. The privacy protection architecture of terminal local encryption, edge ciphertext aggregation, and platform ciphertext-state collaboration constructed by this method can effectively adapt to the hierarchical scenario of IoT terminal-edge-platform, and show strong protection ability for privacy information such as device status and sensitive features in real attack simulations, providing a feasible technical path for the balance between data collaborative training and privacy security in scenarios such as industrial IoT.

To verify the efficiency adaptability of the method in the hierarchical collaborative training of the IoT, the experiment focuses on the time consumption of the two core links of terminal local training encryption of the federated learning algorithm and edge model aggregation of edge computing nodes, and explores the delay characteristics of the method under data flow and privacy protection operations under different numbers of terminal participants and model aggregation rounds. The experimental results are shown in Fig. 2 and Fig. 3, respectively.

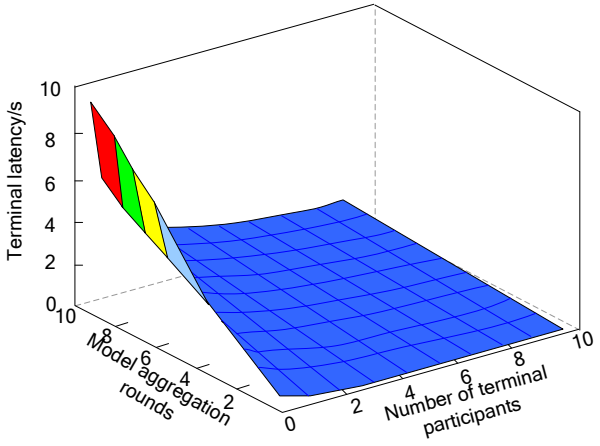

Fig. 2. Local training encryption delay of terminal.

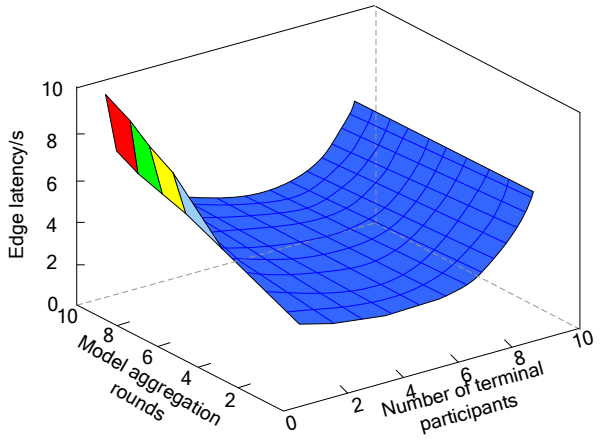

Fig. 3. Edge model aggregation delay.

Fig. 2 presents the law of the influence of the number of terminal participants and the number of model aggregation rounds on the terminal local training encryption delay, and Fig. 3 shows the influence of the two on the edge model aggregation delay. From the terminal side, when the number of terminal participants is stable, the increase in the number of model aggregation rounds significantly increases the delay. This is because each round of aggregation needs to carry out local model training and verifiable dual-key ElGamal encryption in turn. The increase in rounds accumulates the time consumption of training iterations and encryption operations, and also pushes up the network transmission delay; while when the number of model aggregation rounds is stable, the increase in the number of participants disperses the training data volume of a single terminal. Although the number of encryption operations increases, the decrease in training time offsets the increase in encryption time, and the delay shows a downward trend. From the edge side analysis, when the number of model aggregation rounds is stable, the increase in the number of terminal participants continuously increases the edge model aggregation delay and the growth rate accelerates. This is because the edge node executes operations such as ciphertext reception and parameter supplementation for dropped-out participants. When the number of participants is small, the constant optimization can offset the time consumption, and when the number exceeds the threshold, the time consumption of dropped-out supplementation dominated by complexity increases sharply.

Comprehensively, from Fig. 2 and Fig. 3, through terminal local verifiable dual-key ElGamal encryption, edge MPSDGS dropout detection, and MP-Update dynamic weighted aggregation, a hierarchical privacy protection system is built. The terminal side effectively avoids the exposure of sensitive data, and the edge side greatly reduces the risk of ciphertext inference attacks, achieving the rigid protection of privacy in the IoT data collaborative training at a reasonable delay cost; at the same time, the delay growth law adapts to the characteristics of the IoT scenario. The delay on the terminal side decreases with the increase in the number of participants, which is in line with the IoT terminal distributed deployment and data fragmentation processing mode, and has more efficiency advantages in large-scale terminal collaboration. Although the delay on the edge side increases with the scale, it is still within an acceptable range under the typical business scale of the IoT, reflecting the good balance between privacy protection and efficiency adaptability of the method. Compared with the plaintext federated learning without privacy protection and the single encryption weak protection scheme, this method achieves a several-fold increase in privacy protection strength with a controllable delay increment, providing a solution with both security and practicality for data collaborative training in scenarios such as industrial IoT.

To comprehensively verify the actual efficacy of this method in the dimension of privacy protection, the data privacy degree P_{index} index is introduced for quantitative evaluation. The methods studied by Bezanjani et al. [6] and Samriya et al. [7] are selected as the control group. Through a controlled variable experiment, the differences in privacy protection intensity among different schemes are clearly presented. The calculation formula of P_{index} is as follows:

$$P_{index} = \frac{Var(A) - Var(T)}{Var(A)} \tag{10}$$

Among them, A is the original data, and T is the data after applying privacy protection technology for transformation. The higher the value of P_{index}, the greater the difference between the transformed data and the original data, and the better the privacy.

The evaluation results of the data privacy degree P_{index} of the three methods under different numbers of terminal participants are shown in Fig. 4.

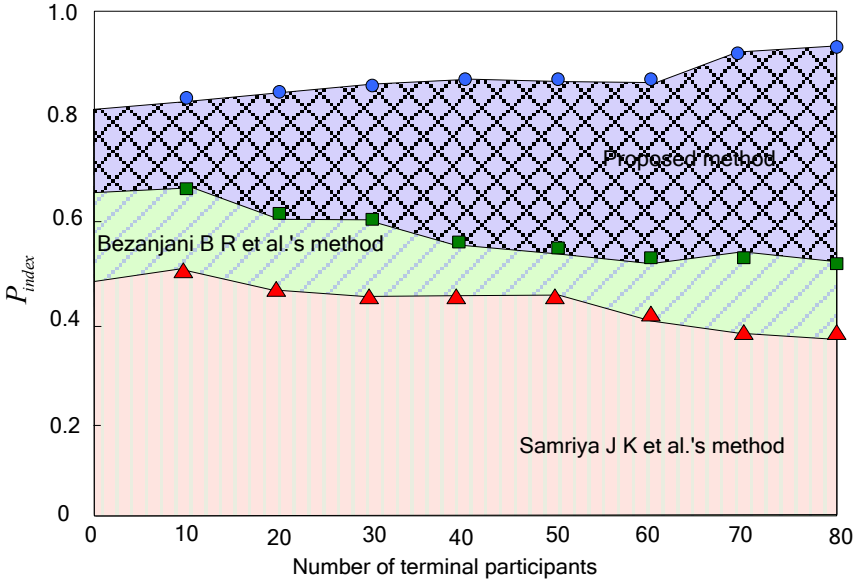

Fig. 4. Evaluation results of data privacy level P_{index} for three methods.

Analyzing the P_{index} situation in Fig. 4, it can be seen that the P_{index} of this method always maintains a high-level interval above 0.8, and shows a small upward trend as the number of terminal participants increases from 0 to 80. This characteristic originates from the architecture constructed in this paper: terminal-side dual-key ElGamal encryption + edge ciphertext aggregation (MP-Update algorithm). On the terminal side, through the asymmetric key mechanism, it ensures that the encrypted sensitive data of the device can only be verified and cannot be broken; on the edge side, relying on the ciphertext aggregation algorithm, the model parameter collaboration is completed without decryption, fundamentally blocking the privacy inference path in the data transmission-aggregation link. Even if the terminal scale expands, the heterogeneous encryption characteristics of multi-device data instead enhance the anti-inference ability of the ciphertext, making P_{index} achieve a positive gain as the number of terminals increases, verifying the robustness of the method in terms of privacy protection in large-scale IoT scenarios. In contrast, although the P_{index} of the Bezanjani method [6] can be maintained in the 0.6-0.7 interval, its single-key design makes the central server a single point of privacy leakage. The larger the terminal scale, the higher the probability of ciphertext being broken, and the privacy protection intensity shows a fluctuating downward trend as the number of terminals increases, making it difficult to adapt to the scenarios of IoT terminal dispersion and dynamic scale. The P_{index} of the Samriya method [7] has been lower than 0.4 for a long time. The root cause is that its complex cryptographic operations and model reasoning have extremely high requirements for the computing power of edge devices. The widespread resource-constrained problem of IoT terminals makes it unable to complete efficient encryption and analysis locally, and it is forced to rely on cloud computing, resulting in an increased risk of privacy leakage during the data transmission-cloud processing process, and the privacy protection efficiency is always at a low level.

As shown in Fig. 4, our method maintains a privacy degree above 0.8 across different participant scales, significantly outperforming Bezanjani's method (0.6-0.7) and Samriya's approach (<0.4). This superiority stems from our dual-key encryption and ciphertext aggregation architecture, which fundamentally blocks privacy inference paths during data transmission and aggregation.

In summary, through the innovative design of hierarchical encryption to decouple cloud-edge dependencies and ciphertext aggregation to block inference paths, the proposed method achieves dual breakthroughs in privacy protection intensity and scenario adaptability. It can balance the needs of data collaboration and the rigidity of privacy protection in complex scenarios where the scale of IoT terminals changes dynamically, providing more reliable privacy and security guarantees for cross-level data flow in various application fields of the IoT.

IV. CONCLUSION

Focusing on the challenges of IoT data privacy protection, this paper proposes a privacy protection method based on the collaborative optimization of edge computing and federated learning. It constructs a hierarchical architecture covering multiple entities, integrates technologies such as homomorphic encryption and dropout supplementation, and achieves significant results through experimental verification. In terms of privacy protection effectiveness, through terminal-local dual-key ElGamal encryption, edge ciphertext aggregation, and platform global secure aggregation, full-process privacy protection of IoT data is realized. In the dimension of efficiency adaptability, the terminal-local training encryption delay increases with the number of aggregation rounds and decreases with the increase in the number of participants. Although the edge model aggregation delay accelerates with scale growth, it remains within an acceptable range under typical IoT business scales, verifying the efficiency resilience of the method and achieving a balance between privacy protection and computational efficiency. Therefore, the proposed method breaks through the limitations of traditional schemes such as cloud dependence, single-point risks, and poor computing power adaptability. By means of hierarchical encryption to decouple cloud-edge dependencies and ciphertext aggregation to block inference paths, it provides reliable privacy guarantees for cross-level data flow in fields such as industrial IoT and smart healthcare, promotes the collaborative development of edge intelligence and privacy computing, and has good engineering application value and scenario expansion potential. Future work can deepen research on lightweight encryption algorithms to further adapt to IoT terminals with extremely limited resources and improve the universality of the method.

The proposed privacy-preserving framework exhibits certain limitations that motivate subsequent research endeavors. First, despite optimizing verifiable dual-key ElGamal encryption, the computational overhead remains prohibitive for ultra-low-power devices (e.g., Class 0 sensors), necessitating exploration of lightweight homomorphic encryption or LWE-based alternatives. Second, current validation is confined to single-domain industrial sensor data; cross-domain applicability across heterogeneous IoT ecosystems (smart healthcare, urban sensing) with divergent data patterns and QoS demands requires rigorous evaluation. Third, the threat model must expand beyond semi-honest adversaries to counter sophisticated attacks like model poisoning or coordinated edge-node compromises. Fourth, ciphertext transmission overhead in large-scale deployments demands communication-efficient strategies such as adaptive aggregation or model compression. Finally, an end-to-end energy consumption analysis on physical IoT hardware is crucial to validate practical sustainability across encrypted training cycles.Future research work will systematically explore the core challenges of IoT security and efficiency: on the one hand, it will focus on innovative lightweight security mechanisms, develop customized homomorphic encryption variants suitable for Class 0 IoT sensors with extremely limited resources, and balance security and computational costs through algorithm tailoring and hardware collaborative design; On the other hand, expanding the boundaries of security verification, building a unified verification framework across heterogeneous scenarios such as intelligent healthcare and

urban sensing, and solving the problems of privacy leakage and permission control in multi domain data fusion.

REFERENCES

[1] A. Dayyani, and M. Abbaspour, "SybilPSIoT: Preventing Sybil attacks in signed social internet of things based on web of trust and smart contract," IET Communications (COM), Vol. 18, no. 3, pp. 258-269, February 2024.

[2] N. Elsakaan, and K. Amroun, "A novel privacy-aware global infrastructure for ecological footprint calculator based on the internet of things and blockchain," The Journal of Supercomputing, vol. 80, no. 7, pp. 9494-9531, May 2024.

[3] R. De, and I. Nanda, "Network/security threats and countermeasures for cloud computing," Acta Electronica Malaysia, vol. 7, no. 1, pp. 1-3, November 2022.

[4] J. Zhang, X. Li, P. Vijayakumar, W. Liang, V. Chang, and B. B. Gupta, "Graph sparsification-based secure federated learning for consumer-driven internet of things," IEEE Transactions on Consumer Electronics, vol. 70, no. 3, pp. 5188-5200, August 2024.

[5] H. Ghasemi, and S. Babaie, "A new intrusion detection system based on SVM-GWO algorithms for internet of things," Wireless Networks, vol. 30, no. 4, pp. 2173-2185, May 2024.

[6] B. R. Bezanjani, S. H. Ghafouri, and R. Gholamrezaei, "Fusion of machine learning and blockchain-based privacy-preserving approach for healthcare data in the internet of things," The Journal of Supercomputing, vol. 80, no. 17, pp. 24975-25003, August 2024.

[7] J. K. Samriya, C. Chakraborty, A. Sharma, M. Kumar, and S. K. Ramakuri, "Adversarial ML-based secured cloud architecture for consumer internet of things of smart healthcare," IEEE Transactions on Consumer Electronics, vol. 70, no. 1, pp. 2058-2065, February 2024.

[8] M. Prakash, and K. Ramesh, "ECAUT: ECC-infused efficient authentication for internet of things systems based on zero-knowledge proof," The Journal of Supercomputing, vol. 80, no. 17, pp. 25640-25667, August 2024.

[9] S. Shree, C. Zhou, and M. Barati, "Data protection in internet of medical things using blockchain and secret sharing method," The Journal of Supercomputing, vol. 80, no. 4, pp. 5108-5135, March 2024.

[10] A. S. Kumar, L. Zhao, and X. Fernando, "Asynchronous federated based vehicular edge computation offloading," IEEE Transactions on Vehicular Technology, vol. 73, no. 12, pp. 19350-19360, December 2024.

[11] A. Khanna, G. Anjali, N. K. Verma, and K. J. Naik, "A GRL-aided federated graph reinforcement learning approach for enhanced file caching in mobile edge computing," Computing, vol. 107, no. 1, pp. 40, December 2024.

[12] A. Rafiq, M. Wei, P. Wang, and D. K. Jain, "Delay aware 6TISCH IIoT networks for energy efficient data transmission by adopting federated learning and edge computing," IEEE Transactions on Consumer Electronics, vol. 70, no. 3, pp. 5911-5928, August. 2024.

[13] A Muthukumar,A Giridhar,G Raghavendra,UV Teja,GS Deepak and GA Kumar.l.Data Security Enhancements in IoT Communication Networks Using Homomorphic Encryption and Location Based Data Access.2025 International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI),pp. 393-399, January 2025.

[14] N. Agarwal, and S. Joshi, "Federated learning-based task offloading in a UAV-Aided cloud computing mobile network," IEEE Transactions on Vehicular Technology, vol. 73, no. 10, pp. 15751-15756, October 2024.

[15] Z. Abou El Houda, H. Moudoud, B. Brik, and L. Khoukhi, "Blockchain-enabled federated learning for enhanced collaborative intrusion detection in vehicular edge computing," IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 7, pp. 7661-7672, July 2024.

[16] O. Manjang, Y. Zhai, J. Shen, J. Tchaye-Kondi, and L. Zhu, "Anchor model-based hybrid hierarchical federated learning with overlap SGD," IEEE Transactions on Mobile Computing, vol. 23, no. 12, pp. 12540-12557, December 2024.

[17] J. Tchaye-Kondi, Y. Zhai, J. Shen, A. Telikani, and L. Zhu, "Adaptive period control for communication efficient and fast convergent federated learning," IEEE Transactions on Mobile Computing, vol. 23, no. 12, pp. 12572-12586, December 2024.

[18] H Park, J Lee.LMSA: A Lightweight Multi-Key Secure Aggregation Framework for Privacy-Preserving Healthcare AIoT.CMES - Computer Modeling in Engineering and Sciences, 2025, 143(1):827-847.Vol. 143, no. 11, pp.827-847,April 2025.

[19] Y. M. Saputra, D. N. Nguyen, D. T. Hoang, Q. -V. Pham, E. Dutkiewicz, and W. -J. Hwang, "Federated learning framework with straggling mitigation and privacy-awareness for AI-based mobile application services," IEEE Transactions on Mobile Computing, vol. 22, no. 9, pp. 5296-5312, September 2023.

[20] H. R. Zang, T. Y. Wang, S. Y. Jiang, and K. Ma, "Centralized federated learning encryption simulation in internet of things environment," Computer Simulation, vol. 42, no. 1, pp. 400-404, January 2025.

[21] A. Alahmadi, H. A. Khan, G. Shafiq, J. Ahmed, B. Ali, M. A. Javed, M. Z. Khan, R. H. Alsisi, and A. H. Alahmadi, "A privacy-preserved IoMT-based mental stress detection framework with federated learning," The Journal of Supercomputing, vol. 80, no. 8, pp. 10255-10274, May 2024.