Alexandria Engineering Journal 118 (2025) 11–20

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Original Article

A privacy-preserving federated learning scheme with homomorphic encryption and edge computing

Bian Zhu *, Ling Niu

School of Computer Science and Technology, Zhoukou Normal University, Zhoukou, Henan province, 466000, China

ARTICLE INFO

Keywords:
Homomorphic encryption
Privacy protection
Federated learning
Edge computing
Secure aggregation
Trust chain

ABSTRACT

With the rapid advancement of data-driven technologies, safeguarding data privacy has become a focal concern in both academia and industry. Traditional data processing methods typically rely on centralized storage and computation, which increases vulnerability to privacy breaches, particularly during data transmission and storage. To address these challenges, we propose a privacy-preserving federated learning framework integrating homomorphic encryption with an added trust chain. The trust chain enables transparent, immutable recording of data processing stages, significantly enhancing system reliability and trustworthiness. Participants employ homomorphic encryption to ensure that data remains encrypted throughout transmission and aggregation, thereby preventing privacy leaks. Additionally, this method leverages edge computing nodes to boost computational efficiency and reduce communication and computation overhead. Specifically, we designed privacy-preserving algorithm modules for participants, edge nodes, and the sensing platform, including local encryption, secure aggregation, and global update algorithms, ensuring robust data security at every stage. Validation on the MNIST dataset demonstrates that our approach surpasses the traditional federated learning algorithm (FedAvg) in performance. Furthermore, under varying key lengths, the method shows reasonable encryption and decryption overhead, with the trust chain further securing each transaction.

1. Introduction

With the rapid development of data-driven technologies, data privacy protection has become an increasingly prominent issue in both academia and industry [1,2]. The vast amount of sensitive information generated and processed daily makes it crucial to protect this data from unauthorized access and potential breaches [3]. Traditional data processing methods typically rely on centralized storage and computation, where data is aggregated on central servers. However, this centralized approach is prone to privacy breaches during data transmission and storage. If the central storage or transmission process is attacked or compromised, sensitive information may be exposed, posing significant risks to individuals and organizations [4–7].

To address these privacy concerns, federated learning has gained increasing attention as a decentralized data processing method [8,9]. Federated learning enables multiple devices or entities to collaboratively train models without sharing raw data, thereby reducing privacy risks to some extent. However, while federated learning mitigates some of the privacy challenges of centralized systems, it still has potential vulnerabilities. Even within a federated system, the transmission of model updates can inadvertently expose information about local datasets [10–12]. This has driven the need for further development of privacy-preserving techniques in federated learning frameworks [13,14].

In this context, homomorphic encryption has emerged as a powerful solution to enhance privacy protection in federated learning. Homomorphic encryption allows computations to be performed on encrypted data without the need for decryption, ensuring that sensitive information remains secure throughout the entire data processing pipeline [15–18]. This capability makes homomorphic encryption an ideal tool for addressing the privacy vulnerabilities in federated learning [19,20].

This paper proposes a privacy-preserving federated learning method based on homomorphic encryption, integrating homomorphic encryption into the federated learning framework. Our approach introduces a key generation center that provides a secure mechanism for distributing encryption keys. Each participant employs homomorphic encryption to ensure that their data remains encrypted throughout transmission and aggregation, preventing potential privacy breaches. Additionally, we incorporate edge computing nodes to improve computational efficiency and reduce system communication and computation overhead.

We design specific privacy-preserving algorithm modules for participants, edge nodes, and the sensing platform, including local encryption,

* Corresponding author.
E-mail address: zhuhaoyun@zknu.edu.cn (B. Zhu).

https://doi.org/10.1016/j.aej.2024.12.070
Received 31 October 2024; Received in revised form 9 December 2024; Accepted 19 December 2024
Available online 17 January 2025
1110-0168/© 2025 The Author(s). Published by Elsevier B.V. on behalf of Faculty of Engineering, Alexandria University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

secure aggregation, and global update algorithms, ensuring data security at each stage of the federated learning process. Using the MNIST dataset for validation, the results show that despite the use of homomorphic encryption, the model's accuracy is almost unaffected and remains comparable to traditional federated learning algorithms such as FedAvg. Furthermore, we evaluate the encryption and decryption time overhead under different key lengths, demonstrating that this method provides reasonable computational efficiency while effectively protecting privacy.

The main contributions of this paper are as follows:

- This paper presents an innovative federated learning framework that leverages homomorphic encryption to provide comprehensive privacy protection for participants' data during both transmission and computation, effectively addressing the privacy risks inherent in traditional federated learning systems.
- By integrating a key generation center, the system ensures a secure key distribution mechanism. Furthermore, the use of edge computing nodes and a distributed architecture with the sensing platform enhances computational efficiency, making it especially well-suited for large-scale, distributed environments.
- The paper introduces privacy-preserving algorithm modules for participants, edge computing nodes, and the sensing platform, all based on homomorphic encryption. These modules include local encryption algorithms as well as secure aggregation and update algorithms, ensuring that data remains encrypted throughout the entire learning process and effectively preventing privacy leaks.

The structure of this paper is as follows: the second section reviews related work, the third section presents the proposed method, the fourth section discusses the experiments, and the final section concludes the paper.

2. Related work

2.1. The current development of mobile crowdsensing

Since its introduction, Mobile Crowdsensing (MCS) has become a popular topic in several research fields. Extensive studies have focused on its applications, task allocation, privacy protection, and incentive mechanisms [21,22]. Significant progress has been made in these areas, laying a solid foundation for the widespread application of MCS systems.

In the realm of task allocation, numerous algorithms and models have been developed to enhance task assignment efficiency. Traditional approaches to task allocation often consider users' geographical locations and the resource capacities of their devices. For instance, Ganti et al. introduced a location-based task allocation model designed to achieve extensive coverage of sensing tasks while minimizing the load on participants [23]. Another category of task allocation methods targets device heterogeneity, investigating strategies to efficiently allocate tasks across devices with varying resource configurations. Guo et al. presented a dynamic task allocation algorithm that accounts for participants' resource constraints, such as battery life and processing power, assigning different tasks to diverse devices to optimize system-wide performance [24]. Recently, artificial intelligence techniques, such as machine learning, have been increasingly integrated into MCS task allocation, leveraging predictive insights into user behavior and environmental variations to further boost allocation efficiency. Privacy protection is another critical focus within MCS research, given that MCS systems rely on participants' mobile devices for data collection and transmission, which inherently raises privacy concerns. Early privacy protection methods predominantly utilized anonymization techniques, such as removing identity information [25]. However, with advances in privacy attack methods, researchers found anonymization alone insufficient to prevent reconstruction attacks using auxiliary data, such as location information. Consequently, more sophisticated privacy-preserving techniques have emerged. Differential privacy, for example, has become a widely adopted solution in MCS, safeguarding user privacy by introducing noise into sensing data. First proposed by Dwork et al. differential privacy demonstrated a framework for balancing strong privacy protection with data utility [26]. Subsequent research has focused on optimizing the balance between privacy level and data accuracy within differential privacy models to make them more applicable to real-time data collection in MCS. Additionally, Multi-Party Computation (MPC) has been employed in MCS privacy preservation [27], enabling multiple participants to collaboratively process computations without disclosing individual data. Homomorphic encryption, pioneered by Gentry et al. presents another innovative privacy solution by allowing computation on encrypted data, thus preventing data exposure during processing [28]. Recent studies indicate that combining homomorphic encryption with differential privacy may offer enhanced privacy protection within MCS systems.

MCS has seen extensive application across diverse domains, showcasing its versatility and impact. One prominent example is traffic monitoring, where MCS has proven highly effective. Zhang et al. introduced a real-time traffic monitoring system leveraging MCS to gather data from drivers' mobile devices, providing timely alerts on road congestion and traffic incidents [29]. Similarly, MCS has been employed in environmental monitoring initiatives, such as the air quality monitoring system designed by Sheng et al. which utilizes smartphone sensors to detect urban air pollutant levels, thereby contributing to urban management and environmental conservation efforts [30]. Public safety is another area where MCS demonstrates substantial value, enabling the creation of distributed monitoring and alert systems. Through MCS, participants can report incidents such as natural disasters, fires, and traffic accidents in real-time, greatly enhancing emergency response capabilities. Additionally, MCS has significant applications in health monitoring. For instance, smartphones and wearable devices can be used to gather health-related data, such as step counts and heart rates, allowing for both individual health management and large-scale public health data analysis. This capability enables early disease outbreak detection and offers a framework for preventive healthcare measures [31].

2.2. Research status on privacy protection in mobile crowdsensing

The widespread adoption of Mobile Crowdsensing (MCS) has brought data privacy concerns to the forefront of research, as participants' mobile devices often collect highly sensitive information, including location and behavioral data. Ensuring the privacy of this data has thus become a central challenge, prompting researchers to explore a range of privacy-preserving solutions, from encryption and anonymization techniques to data perturbation and trust frameworks [32]. One notable solution, proposed by Castelluccia et al. is a homomorphic encryption-based scheme that enables computations to be performed on encrypted data, maintaining data privacy throughout the process [33]. This method's key advantage lies in its ability to carry out operations without decrypting the data, allowing MCS systems to process data securely without exposing sensitive information. The theoretical security of this approach has been validated, demonstrating its effectiveness in mitigating privacy risks in real-world applications [34]. Alsheikh et al. addressed the balance between privacy protection and data accuracy by introducing an incentive mechanism that optimizes both aspects [35]. Their approach allows participants to receive privacy safeguards and rewards while contributing high-quality sensing data. By implementing a carefully designed incentive model, this scheme encourages participants to provide accurate data, reinforcing both data quality and personal privacy protection.

To enhance data reliability, Zhao et al. developed a data reliability estimation model leveraging cryptographic techniques, specifically utilizing zero-knowledge proof [36]. This model allows participants to verify the authenticity and reliability of their data without revealing the

actual data content, ensuring high-quality data contributions within the Mobile Crowdsensing (MCS) system. The application of zero-knowledge proof significantly improves the system's resilience to untrusted participants, minimizes the influence of false data, and safeguards participant privacy [37]. Wang et al. introduced a privacy-preserving approach that integrates data perturbation with a trust-based framework. In this model, data perturbation obfuscates participants' real data by adding noise, offering robust privacy protection. Complementing this, the trust framework evaluates both participant behavior and data integrity, establishing a trust scoring mechanism to distinguish reliable participants. This combined approach strikes a balance between privacy preservation and data reliability, maintaining the overall efficacy of the MCS system [38]. Additionally, Sucasas et al. combined cryptographic and anonymization techniques to propose a comprehensive privacy protection solution [39]. Cryptographic methods ensure the security of data during transmission and storage, while anonymization techniques enhance participant privacy by removing or obfuscating personally identifiable information. This approach effectively reduces the likelihood of privacy leaks while ensuring that the system can still collect high-quality sensing data [40]. Researchers have proposed various innovative solutions to address privacy protection in MCS, incorporating cryptographic techniques, homomorphic encryption, zero-knowledge proofs, and exploring how incentive mechanisms and trust frameworks can maximize data accuracy and quality while safeguarding privacy. These studies provide a solid theoretical and practical foundation for privacy protection in future MCS systems. However, further optimization and improvements are needed to address the privacy challenges in large-scale applications [41].

3. Method

In this paper, we introduce a trusted key generation center into the edge computing network architecture and propose a privacy-preserving federated learning system model based on homomorphic encryption. The key generation center generates and distributes encryption keys to participants, edge computing nodes, and the sensing platform, ensuring that all data remains encrypted during transmission and processing to safeguard privacy. Additionally, we design privacy-preserving algorithms based on homomorphic encryption for the three main entities in the system: participants, edge computing nodes, and the sensing platform. These include the local encryption algorithm for participants, the secure aggregation and update algorithm at the edge, and the global secure aggregation and update algorithm. These algorithmic modules enable encrypted processing and secure aggregation of private data throughout various computation stages, ensuring that data privacy is protected at every step while maintaining the correctness and efficiency of federated learning computations.

3.1. System model

This paper proposes a privacy-preserving federated learning method based on homomorphic encryption. As shown in Fig. 1, the system architecture consists of five main entities: the requester, participants, edge computing nodes, the sensing platform, and the key generation center. Below is a description of the functions of each entity and their roles within the system.

Requester: The requester is the entity that initiates the sensing task, usually a government agency, enterprise, or research institution with data needs. The requester submits a sensing task request to the system, including details such as the task content, data requirements, and target accuracy. The requester is also responsible for receiving the final aggregated results for analysis and decision-making.

Participants: Participants are the executors of the sensing tasks, typically ordinary users equipped with mobile devices. These devices collect the required sensing data through built-in sensors, such as GPS, accelerometers, and microphones. In this approach, participants first decrypt and update their local model parameters using the private key distributed by the key generation center. After completing the local training, they encrypt the updated model parameters using the Paillier encryption scheme, a form of homomorphic encryption. The encrypted data is then uploaded to the edge computing node, ensuring privacy protection during transmission.

Edge Computing Nodes:Participants are the executors of the sensing tasks, typically ordinary users equipped with mobile devices. These devices collect the required sensing data through built-in sensors, such as GPS, accelerometers, and microphones. In this approach, participants first decrypt and update their local model parameters using the private key distributed by the key generation center. After completing the local training, they encrypt the updated model parameters using the Paillier encryption scheme, a form of homomorphic encryption. The encrypted data is then uploaded to the edge computing node, ensuring privacy protection during transmission.

Sensing Platform: The sensing platform serves as the system's central data processing hub, typically hosted in the cloud. It gathers encrypted intermediate results from various edge computing nodes and carries out global model aggregation. By keeping the data encrypted throughout the entire process, the platform avoids direct access to participants' raw data, effectively minimizing privacy breach risks. Upon completing the global aggregation in encrypted form, the platform returns the aggregated results to the requester.

Key Generation Center: The key generation center is tasked with generating and distributing key pairs to all system participants, producing the public and private keys required for encryption and decryption operations. The security and reliability of this center form the cornerstone of the system's privacy protection framework, ensuring that only authorized participants can securely encrypt and decrypt their data.

In this approach, participants employ homomorphic encryption to maintain data privacy and security during transmission and processing. Edge computing nodes and the sensing platform then work together to execute secure aggregation operations on the encrypted data. This architecture not only robustly protects participants' privacy but also enhances computational efficiency and response times, making it ideal for large-scale, distributed mobile crowdsensing applications.

3.1.1. Threat model and design goals

In the privacy-preserving federated learning system model based on homomorphic encryption, we comprehensively consider the trustworthiness of each entity in the system (including participants, edge computing nodes, the sensing platform, the requester, and the key generation center) as well as the potential security threats they may face. Based on the behavior assumptions of these entities, the system needs to ensure the correctness of data processing and computation while protecting privacy.

Participants: In this system model, participants are assumed to be trusted. They follow the protocol to collect sensing data and perform local model training and updates. Participants do not intentionally tamper with data or the training process, nor do they attempt to compromise the system or leak the private data of other participants. Although participants' devices may have resource limitations (e.g., computational power and battery life), they strictly adhere to the encryption standards provided by the key generation center when encrypting local model parameters using homomorphic encryption. Even so, participants' private data must still be encrypted to prevent theft or leakage during transmission to edge computing nodes or the sensing platform.

Edge Computing Nodes: Edge computing nodes are assumed to be semi-honest, meaning that while they follow the protocol to perform data processing and aggregation, they may attempt to infer sensitive information by analyzing the encrypted data uploaded by participants. In this case, edge computing nodes do not maliciously alter data or refuse service, but they may passively try to infer participants' private information. Therefore, edge computing nodes must perform initial data aggregation in ciphertext form to ensure that no intermediate

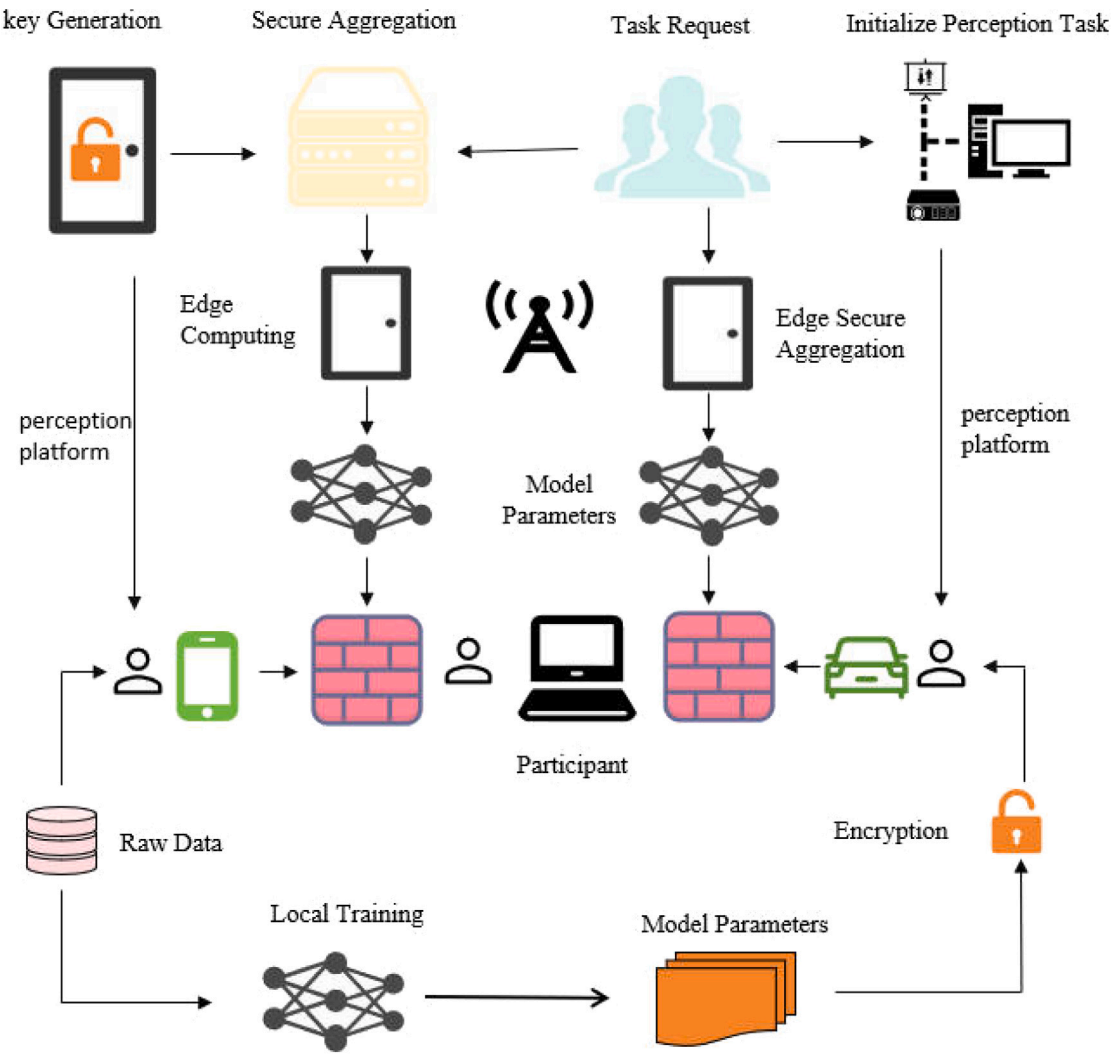

Fig. 1. Federated learning privacy protection system model.

results reveal participants' private data.

Sensing Platform: The sensing platform is assumed to be a semi-honest entity, meaning it adheres to the protocol for global model aggregation and returns the result to the requester. However, it may attempt to analyze the encrypted data during the aggregation process in an effort to infer participants' private information. To mitigate this potential threat, the system is designed with homomorphic encryption, ensuring that all computations are performed on encrypted data. This approach effectively prevents the sensing platform from accessing or decrypting participants' raw data at any stage.

Requester: The requester of the sensing task may be an untrusted entity. While the requester receives the final aggregated result, they could attempt to infer participants' private information by analyzing the returned model. Since the requester does not directly participate in data processing, the system must ensure that the final model provided to the requester fulfills the task requirements without disclosing any private information about individual participants.

Key Generation Center: The key generation center is considered a trusted entity. It is responsible for generating and distributing encryption and decryption key pairs for participants, edge computing nodes, and the sensing platform. The security of the key generation center is the core of the entire system's privacy protection mechanism, and it is assumed that it does not maliciously distribute incorrect keys or attempt to compromise participants' data privacy. The key generation center ensures that each entity can correctly encrypt and decrypt data, guaranteeing the overall security and privacy protection capabilities of the system.

3.1.2. System workflow

For the proposed privacy-preserving federated learning method based on homomorphic encryption, the main design goals are as follows:

- **Privacy Protection**: Ensures that participants' private data remains encrypted throughout the execution of the sensing task, preventing data leakage via homomorphic encryption.
- **Efficient Computation**: Performs ciphertext computations on edge computing nodes and the sensing platform, ensuring both the efficiency and accuracy of data processing.
- **Secure Aggregation**: Ensures the security of data during the aggregation process at the edge computing nodes and the sensing platform, preventing intermediate data leakage.
- **Key Management**: The key generation center provides the necessary public and private keys for participants, edge nodes, and the sensing platform, ensuring secure data encryption and decryption.

The specific execution process of the system is shown in Fig. 2. The figure illustrates how the requester, participants, edge computing nodes, and the sensing platform, under the control of the key generation center, complete the processing of encrypted data. The detailed steps are described below:

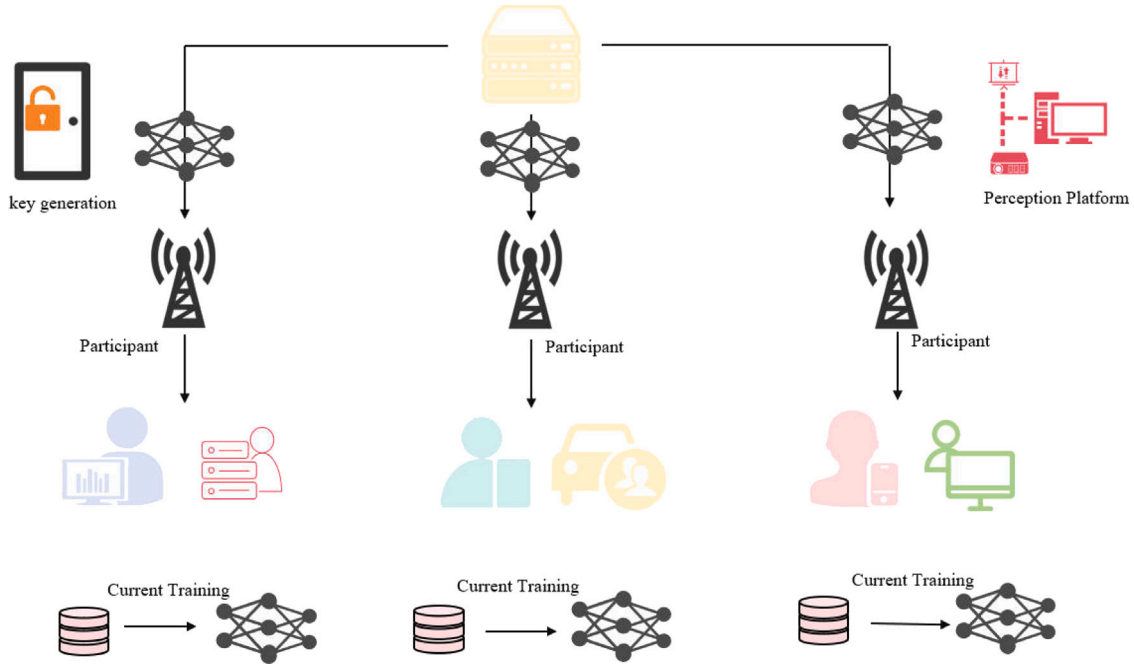

Fig. 2. Federated learning privacy protection system workflow based on homomorphic encryption.

- **Step 1: Sensing Task Initialization** The requester initiates the sensing task by defining the task content, data requirements, and target performance. This includes specific goals, such as collecting environmental data or traffic information from a certain geographic area.
- **Step 2: Task Request and Key Generation** The requester sends the sensing task request to the sensing platform. Meanwhile, the key generation center generates and distributes public and private keys to each participant and edge computing node, ensuring secure encryption and decryption operations in the subsequent steps.
- **Step 3: Task Distribution and Key Assignment** The sensing platform distributes the sensing task and the related public keys to participants. After receiving the task, participants prepare to perform local data sensing and model training.
- **Step 4: Local Model Training and Encryption** After receiving the sensing task, participants use their local devices to collect raw data and perform local model training. Upon completion of training, participants use the public key provided by the key generation center to homomorphically encrypt the local model parameters and upload the encrypted parameters to the edge computing node.
- **Step 5: Secure Aggregation at the Edge** After receiving the encrypted model parameters from multiple participants, the edge computing node performs an initial aggregation of these encrypted parameters in ciphertext form. Since the data remains encrypted, the edge node cannot access the raw data during the computation, ensuring participants' privacy.
- **Step 6: Global Secure Aggregation** Once the edge computing node completes the initial aggregation, it transmits the aggregated results to the sensing platform. The sensing platform further performs global secure aggregation on the encrypted intermediate results. Throughout the process, the sensing platform is unable to access or decrypt participants' raw data, thus protecting data privacy.

Through the above steps, this method achieves privacy protection and secure aggregation in a federated learning environment, meeting the system design goals while ensuring the security and efficiency of data at each stage.

3.2. Homomorphic encryption-based privacy-preserving algorithm

The privacy-preserving algorithm based on homomorphic encryption (Fed_Paillier) proposed in this paper is composed of three main components: the local encryption algorithm for participants, the secure aggregation and update algorithm at the edge, and the global secure aggregation and update algorithm. Together, these components ensure that participants' private data remains protected and is not exposed at any stage of transmission or computation throughout the federated learning process.

3.2.1. Local encryption algorithm of participants

In each round of federated learning, participants first train their local models using local data, generating the updated model parameters ω_i^t. To protect the privacy of these model parameters during upload to the edge computing nodes, participants encrypt them. This paper employs the Paillier encryption scheme, which enables addition operations to be performed on the encrypted data without the need for decryption.

Let N be the modulus generated by the key generation center. The participant uses the public key $pk = (N, g)$ to encrypt the local model parameters ω_i^t. The encryption formula is:

$$Enc(\omega_i^t) = g^{\omega_i^t} \cdot r^N \mod N^2 \tag{1}$$

where g is the base in the public key, and r is a randomly chosen number by the participant to ensure randomness and uniqueness in encryption.

The encrypted parameters $Enc(\omega_i^t)$ are uploaded to the edge computing node, ensuring that the local data and model parameters remain encrypted during transmission, thus protecting data privacy.

3.2.2. Secure aggregation and update algorithm at the edge

Upon receiving the encrypted model parameters uploaded by multiple participants, the edge computing node is responsible for performing aggregation on the model parameters in ciphertext form. Due to the homomorphic addition property of the Paillier encryption scheme, the edge computing node can directly perform addition operations on the encrypted model parameters, achieving secure aggregation of the model parameters from multiple participants.

Suppose there are n participants who have uploaded encrypted local model parameters $Enc(\omega_i^t)$. The edge node performs the following addition aggregation on these encrypted model parameters:

$$Enc(\omega_{agg}^t) = \prod_{i=1}^{n} Enc(\omega_i^t) \mod N^2 \quad (2)$$

Due to the homomorphic addition property of the Paillier encryption scheme, the aggregated result $Enc(\omega_{agg}^t)$ remains in ciphertext form, corresponding to the sum of all participants' model parameters. Once aggregation is completed, the edge computing node uploads the encrypted aggregation result to the sensing platform for further global aggregation.

3.2.3. Global secure aggregation and update algorithm

The sensing platform receives the aggregation results from multiple edge computing nodes and is responsible for completing the global secure aggregation. Similar to the aggregation process at the edge nodes, the sensing platform does not need to decrypt the encrypted model parameters from participants but instead completes global aggregation directly in ciphertext form using the homomorphic encryption property.

Suppose the sensing platform receives the encrypted aggregation results $Enc(\omega_{agg,j}^t)$ from multiple edge nodes, where j represents different edge nodes. The formula for global aggregation is:

$$Enc(\omega_{global}^t) = \prod_{j=1}^{m} Enc(\omega_{agg,j}^t) \mod N^2 \quad (3)$$

This formula ensures that the aggregation results from all edge nodes are further encrypted and aggregated, generating the global model parameters $Enc(\omega_{global}^t)$.

After the global aggregation is completed, the sensing platform sends the encrypted global model parameters back to the participants. Participants use their private key sk to decrypt the global model parameters and update their local model.

$$\omega_{global}^t = L\left((Enc(\omega_{global}^t))^{\lambda} \mod N^2\right) \cdot \mu \mod N \quad (4)$$

where λ and μ are parameters from the participant's private key, and the function $L(u) = \frac{u-1}{N}$. Through decryption, participants obtain the global model parameters and use them for the next round of local model updates.

4. Experiments

4.1. Dataset

In this experiment, we used the MNIST standard dataset [42], which is commonly applied in the field of machine learning. The MNIST dataset [42] consists of 70,000 handwritten digit images (digits 0 to 9), with 60,000 images used for training and 10,000 for testing. Each image is a 28×28 grayscale image, and every image is associated with a corresponding digit label. To simulate participant behavior in a federated learning scenario, we assume that each participant only possesses samples of two categories of handwritten digit images. This data distribution simulates a real-world scenario where participants may only have access to specific types of data samples. Specifically, the MNIST dataset is split by category and randomly assigned to participants, such that each participant's dataset contains only two distinct categories of digit images. This Non-IID (Non-Independent and Identically Distributed) data split reflects the distributed nature of data in practical applications and helps evaluate the performance of the homomorphic encryption-based federated learning algorithm when dealing with non-uniform data. Each edge computing node is responsible for aggregating data from multiple participants. In the experiment, the encrypted data from multiple participants is randomly assigned to each edge computing node, ensuring that the task at each node exhibits a different data distribution.

4.2. Experimental setup

To simulate a privacy-preserving federated learning system based on homomorphic encryption in an edge computing network environment, we implemented a three-layer architecture using Python. The architecture is divided into the user layer, the edge computing layer, and the sensing platform layer. The user layer represents the participants, responsible for collecting local data and conducting local model training. The edge computing layer consists of edge nodes responsible for receiving encrypted model parameters from the user layer and performing secure aggregation operations on encrypted data. The sensing platform layer is responsible for collecting aggregated encrypted data from the edge computing nodes and performing global aggregation operations. To ensure privacy protection, all data transmissions between these layers are encrypted using homomorphic encryption, ensuring that participants' private data remains secure throughout the transmission and aggregation process.

In the experiment, we used the PyTorch (V1.0.0) framework to handle model training and updates, while the Paillier encryption algorithm from the 'phe' package was used to implement homomorphic encryption. Specifically, the key generation center uses Paillier encryption to generate a global private key 'global_private_key' and a global public key 'global_public_key'. The public key is distributed to participants and edge computing nodes. Once the local model training is completed, participants encrypt their local model parameters using the public key before uploading them to the edge computing nodes. The edge nodes then perform secure aggregation on the encrypted model parameters without decrypting them. To maintain privacy, the private key sk is securely stored by the key generation center and is only used for decrypting the final global model. This ensures that data remains encrypted throughout the computations performed by the edge nodes and the sensing platform.

4.3. Experimental parameter settings

In the experiment, we configured the parameters of the federated learning system to ensure the validity of the experiment and optimize system performance. As shown in Table 1, the settings are as follows: First, the number of global aggregation rounds is set to 20, meaning that the sensing platform performs 20 global model aggregation operations throughout the training process. After each aggregation, the global model is distributed to the edge computing nodes for further updates. Additionally, the number of edge aggregation rounds is set to 5, meaning each edge computing node first performs 5 secure aggregation rounds on the models uploaded by local participants before each global aggregation. This ensures that the data from participants is fully aggregated at the edge nodes. At the local participant level, the number of local iterations is set to 20, meaning each participant performs 20 gradient updates on their local model using their local data during training. The batch size is set to 20, ensuring that a small, randomly selected subset of the participant's local dataset is used for each training round, thereby enhancing training efficiency. For model optimization, the learning rate, denoted as η, is set to 0.01, representing the step size for updating model parameters during each iteration. To prevent premature convergence and improve training stability, we introduced a learning rate decay factor of 0.995, which is applied after each global aggregation round. Additionally, the momentum for the SGD optimizer is set to 0.9 to help the model avoid local optima during the optimization process. Finally, for security purposes, we used a 512-bit Paillier encryption key length (n_{length}) to ensure the strength of homomorphic encryption during data transmission and aggregation, safeguarding participants' privacy against potential attacks.

Table 1
Experimental parameter settings.

Parameter name	Default value
Global Aggregation Rounds T/t_{t_2}	20
Edge Aggregation Rounds t_2	5
Local Iterations t_1	20
Batch Size W	20
Learning Rate η	0.01
Learning Rate Decay	0.995
Decay Frequency	1
SGD Momentum	0.9
Security Parameter n_{length}	512

4.4. Evaluation metrics

To verify the effectiveness of the proposed privacy-preserving federated learning method based on homomorphic encryption, the following evaluation metrics are used: accuracy, loss function value, time overhead, and encryption/decryption time for participants. The definitions and mathematical expressions for each evaluation metric are as follows:

The accuracy of the model is used to evaluate the performance of the classification model on the test set. It is defined as the ratio of correctly classified samples to the total number of samples, and its formula is:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} I(\hat{y}_i = y_i) \tag{5}$$

where N denotes the total number of samples in the test set, $\hat{y}_i$ represents the predicted label of the ith sample, and y_i is the true label of the ith sample. The indicator function I equals 1 if $\hat{y}_i = y_i$, and 0 otherwise.

The loss function measures the discrepancy between the predicted values and the true labels. In this experiment, the cross-entropy loss function is used, and its calculation formula is:

$$L(\hat{y}, y) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C} y_{i,k}\log(\hat{y}_{i,k}) \tag{6}$$

where N is the total number of samples, C is the number of classes, $y_{i,k}$ represents the one-hot encoded true label for sample i, and $\hat{y}_{i,k}$ is the predicted probability of sample i belonging to class k.

Time overhead is used to evaluate the computational and communication cost of the system, including model training time, encryption time, decryption time, as well as the edge and global aggregation times. The total time overhead T_{total} is the sum of these components and is expressed as:

$$T_{\text{total}} = T_{\text{train}} + T_{\text{encrypt}} + T_{\text{decrypt}} + T_{\text{edge\_agg}} + T_{\text{global\_agg}} \tag{7}$$

where T_{train} is the time for local model training, T_{encrypt} is the time for encrypting model parameters, T_{decrypt} is the decryption time, $T_{\text{edge\_agg}}$ is the aggregation time at the edge computing nodes, and $T_{\text{global\_agg}}$ is the global aggregation time at the sensing platform.

The encryption and decryption time for participants evaluates the efficiency of homomorphic encryption in a distributed environment. The encryption time T_{encrypt} and decryption time T_{decrypt} are defined as the time taken by participants to encrypt and decrypt their local model parameters during each training round, expressed as:

$$T_{\text{encrypt}} = \sum_{i=1}^{N} T_{\text{encrypt}_i}, \quad T_{\text{decrypt}} = \sum_{i=1}^{N} T_{\text{decrypt}_i} \tag{8}$$

where T_{encrypt_i} and T_{decrypt_i} represent the encryption and decryption time for participant i, respectively.

4.5. Results

4.5.1. Impact of key size on time overhead

To evaluate the impact of different key lengths on system time overhead, this paper adopts three commonly used Paillier encryption key lengths (512 bits, 1024 bits, and 2048 bits) and compares the encryption and decryption time overhead for participants under these three key sizes. Fig. 3 presents the comparison results of the average encryption time and average decryption time for participants with different key sizes. As shown in Fig. 3, the encryption and decryption times for participants increase significantly as the key length increases. When the key length is 512 bits, the average encryption time for participants is relatively short, requiring only a small amount of computation time to complete the encryption operation. This is because shorter key lengths correspond to a smaller key space, resulting in lower computational complexity for encryption. However, when the key length increases to 1024 bits, the average encryption time for participants increases significantly. This is due to the increased computational complexity brought about by the longer key length, which makes the large number operations in the encryption process more time-consuming. When the key length is further increased to 2048 bits, the average encryption time reaches 231.09 s, showing a significant growth compared to the encryption time for 512-bit keys. The trend of decryption time is similar to that of encryption time. When the key length is 512 bits, the decryption time is relatively short, allowing participants to quickly complete the decryption of the global aggregated model. However, when the key length increases to 2048 bits, the decryption time reaches 67.73 s, showing a notable increase compared to the decryption times under 512-bit and 1024-bit keys. This indicates that as the key length increases, the decryption process is similarly affected by the expansion of the key space, requiring participants to perform more complex large number operations, thereby increasing the time overhead for decryption.

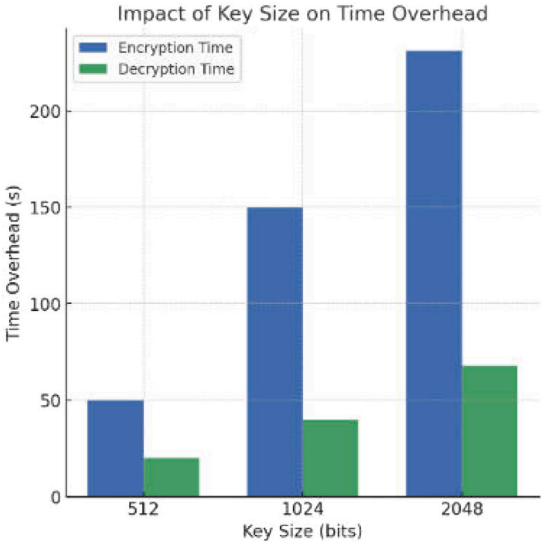

Fig. 3. Time cost of different key sizes.

4.5.2. Impact of the number of participants on encryption and decryption time

To analyze the impact of the number of participants on encryption and decryption times, this paper evaluates the encryption and decryption times for different numbers of participants by changing the randomly selected proportion value ρ. Specifically, the total number of participants K is set to 50, with ρ ranging from $\rho \in [0.2, 1]$, i.e., the number of participants increases from 10 to 50, to assess the impact of different numbers of participants on encryption and decryption times. As shown in Fig. 4, both the average encryption time and decryption time of participants exhibit a linear growth trend as the number of participants increases. When the number of participants is small (e.g., 10 participants), the time spent on encryption and decryption

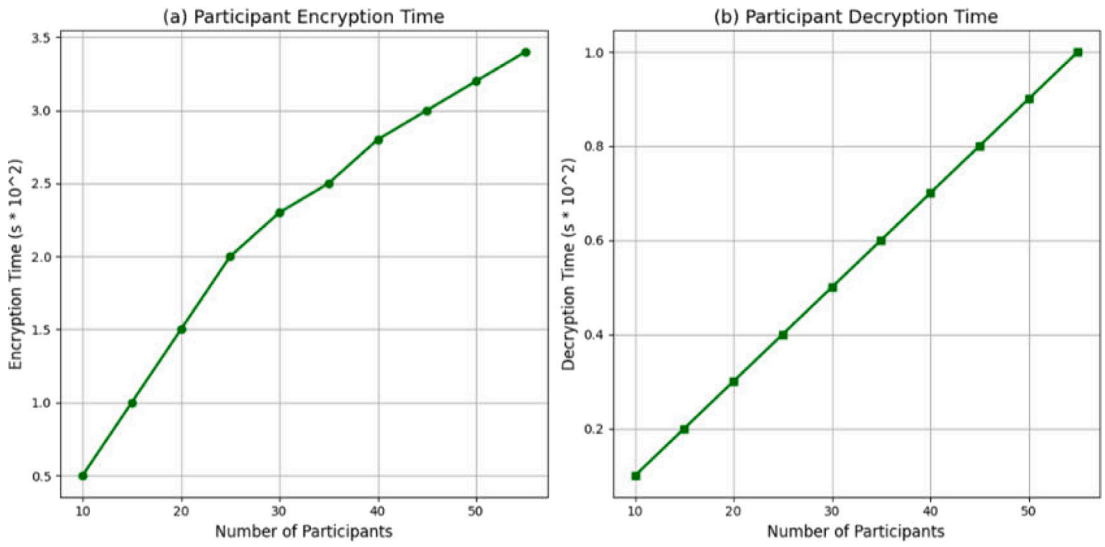

Fig. 4. The encryption and decryption times for different numbers of participants.

is relatively short; however, as the number of participants gradually increases to 50, both encryption and decryption times increase significantly. In detail, Fig. 4(a) shows the change in encryption time with the increase in the number of participants. As the number of participants increases, each participant's encryption time overhead grows because the number of model parameters uploaded by participants increases during the local encryption process. Additionally, due to the inherently high computational complexity of homomorphic encryption schemes, the increase in the number of participants further exacerbates this computational burden, resulting in a noticeable growth in encryption time. Fig. 4(b) illustrates the changes in decryption time. As the number of participants increases, the amount of data to be decrypted after receiving the global model also increases, leading to a gradual rise in decryption time. Similar to encryption operations, the complexity of large-number computations in decryption increases as the number of participants grows, causing decryption time to linearly increase with the number of participants.

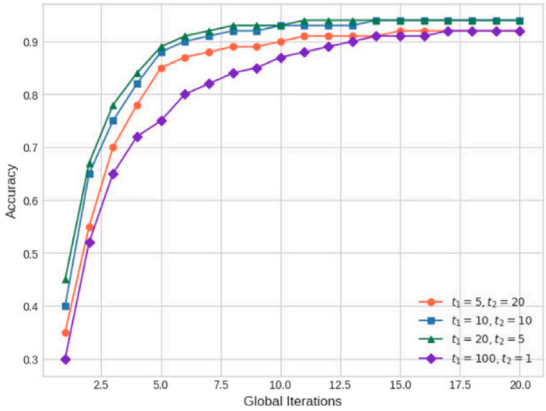

Fig. 5. Accuracy of different iterations.

4.5.3. The impact of iteration allocation on model accuracy

In this experiment, it is assumed that there are 50 participants, 5 edge computing nodes, and 1 sensing platform in the system. Each edge computing node is assigned 10 participants. During each iteration, the sensing platform is responsible for sending global model parameters to the 5 edge computing nodes, which then distribute these parameters to their respective participants for local model training and updating. In this experiment, the total number of global iterations is fixed at 20, and we investigate the impact of different local iteration counts t_1 and edge aggregation counts t_2 on model accuracy.

The Fig. 5 shows the trend of model inference accuracy under four different combinations of t_1 and t_2: $t_1 = 5, t_2 = 20$, $t_1 = 10, t_2 = 10$, $t_1 = 20, t_2 = 5$, and $t_1 = 100, t_2 = 1$. It can be observed that different combinations of local iteration counts and edge aggregation counts have a significant impact on the convergence speed and final accuracy of the model. From the figure, we can observe that when t_1 is small and t_2 is large (e.g., $t_1 = 5, t_2 = 20$), the model converges faster, but the final accuracy is slightly lower than other configurations. This is because fewer local iterations result in smaller updates to the model in each round of training, while frequent edge aggregation operations help the model reach convergence more quickly. Conversely, when t_1 is large and t_2 is small (e.g., $t_1 = 100, t_2 = 1$), the model converges more slowly, but after multiple local training iterations, the final accuracy of the model is higher. This indicates that more local iterations help the model optimize better on the participants' local data, but fewer edge aggregations reduce the frequency of global updates, which slows the overall convergence speed of the model. The experiment demonstrates that the reasonable configuration of t_1 and t_2 can influence both the convergence speed and accuracy of the model. Increasing the number of local iterations helps improve the final accuracy of the model, while increasing the edge aggregation count can speed up the model's convergence.

Therefore, in practical applications, it is necessary to choose the number of local training iterations and edge aggregations based on the system's computational resources and task requirements to find the optimal balance between training speed and model accuracy.

4.5.4. Comparison of accuracy and loss for different methods

To validate the effectiveness of incorporating homomorphic encryption into the federated learning process, this paper compares the proposed privacy-preserving federated learning algorithm based on homomorphic encryption (Fed_Paillier) with the classical federated learning algorithm FedAvg. By analyzing the trends in model accuracy and training loss, the impact of homomorphic encryption on model performance, while ensuring privacy protection, is evaluated.

As shown in Fig. 6, the performance of Fed_Paillier and FedAvg in terms of model accuracy is very close. Throughout the iteration process, the accuracy of both algorithms gradually increases with the number of iterations, ultimately converging at a high accuracy level close to 1.0. The accuracy difference between Fed_Paillier and FedAvg is minimal, indicating that even with the introduction of homomorphic

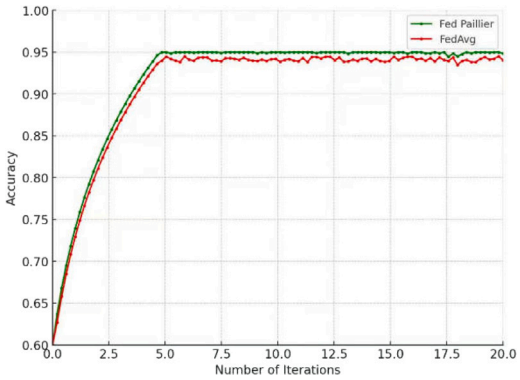

Fig. 6. Accuracy of different methods.

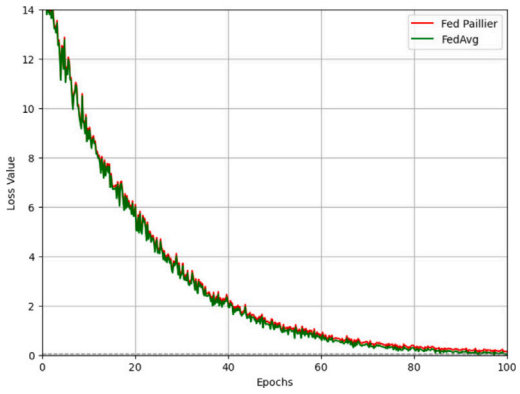

Fig. 7. Comparison of different loss functions.

encryption, the classification performance of the model after global aggregation is almost equivalent to that of the traditional FedAvg algorithm. This verifies that the proposed homomorphic encryption algorithm can maintain the performance of the federated learning model while ensuring data privacy. The encrypted aggregation in each iteration of the Fed_Paillier algorithm did not significantly reduce the model's accuracy, indicating that homomorphic encryption has little impact on the classification ability of the federated learning model, and the model can still effectively learn and optimize participant data.

As shown in Fig. 7, the performance of the Fed_Paillier and FedAvg algorithms in terms of training loss is quite similar. During the initial phase of training, both algorithms exhibit a rapid decrease in loss, and as the number of iterations increases, the training loss gradually stabilizes, approaching a steady value after a sufficient number of iterations. The convergence trend of Fed_Paillier's loss closely mirrors that of FedAvg, with only minimal differences in the final loss values. This indicates that the homomorphic encryption mechanism does not significantly hinder the convergence of training loss in privacy-preserving federated learning. Fed_Paillier is able to effectively perform model aggregation while maintaining the encrypted state, ensuring that training performance remains strong and the model loss converges quickly, even within an encrypted environment.

4.6. Limitations and future work

The shortcomings of this paper mainly lie in two aspects. First, the trust issue of the key generation center introduces a potential single point of failure risk for the system. The paper assumes that the key generation center is fully trustworthy, but in practical applications, over-reliance on a single trusted entity may introduce vulnerabilities. If the key generation center is attacked or compromised, the entire privacy protection mechanism could be at risk. Second, the challenges of model updates and data heterogeneity are also significant. Participants often have widely varying data distributions, which can lead to suboptimal global model updates. In scenarios with strong data heterogeneity, ensuring model convergence and improving overall learning performance remain areas that require further exploration.

To address these shortcomings, several future research directions can be considered. To mitigate the single point of failure of the key generation center, decentralized key management mechanisms, such as blockchain or multi-party computation technologies, could be introduced to enhance the security and robustness of the system. Additionally, to address the issue of data heterogeneity, privacy-preserving personalized models could be explored, allowing participants to optimize locally based on their own data characteristics, while still collaborating on training the global model to improve overall performance.

Privacy-preserving federated learning systems can be extended to various real-world application scenarios to further validate their feasibility and effectiveness. For instance, in the intelligent transportation sector, data from various traffic sensors, cameras, and vehicles is highly sensitive. Federated learning can collaboratively train traffic flow prediction, road congestion monitoring, and autonomous driving algorithms while ensuring data privacy, thus enhancing the intelligence of transportation systems. In the healthcare industry, patient health data is extremely sensitive. Federated learning allows different hospitals to share data features and collaboratively train disease diagnosis and treatment models without disclosing patient privacy, thereby improving diagnostic accuracy and personalized treatment outcomes. In the financial sector, banks and financial institutions can use federated learning to jointly train credit assessment and fraud detection models, avoiding the centralized storage or leakage of sensitive financial data while safeguarding customer privacy and enhancing risk control capabilities. Additionally, in the smart home and IoT domains, home devices can locally train personalized models (such as voice recognition, behavior prediction, etc.) without uploading user data to the cloud, thus protecting user privacy. Through the expansion of these application scenarios, privacy-preserving federated learning systems can not only be validated across different fields but also promote the widespread adoption and development of data privacy protection technologies in more industries. At the same time, they provide valuable experience and technical support to address practical challenges such as data heterogeneity and limited computational resources.

5. Conclusion

This paper presents a privacy-preserving federated learning system based on homomorphic encryption, incorporating a key generation center and a distributed computing framework to protect participants' data privacy throughout both transmission and computation. The system utilizes edge computing nodes and a central sensing platform to securely aggregate encrypted data, enhancing computational efficiency while ensuring robust data protection. Specifically designed for large-scale, distributed mobile crowdsensing applications, the proposed algorithms ensure that data remains encrypted throughout the learning process, effectively preventing data leakage.

In conclusion, this paper offers a practical solution for privacy-preserving federated learning by leveraging homomorphic encryption during both data transmission and processing. The proposed system not only secures participant privacy but also improves computational efficiency, providing valuable insights into the integration of privacy protection with distributed computing. This approach lays a solid foundation for extending privacy-preserving federated learning to broader application domains.

CRediT authorship contribution statement

Bian Zhu: Writing – original draft, Methodology, Formal analysis, Conceptualization. **Ling Niu:** Writing – review & editing, Validation, Investigation.

Declaration of competing interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

[1] H. Fang, Q. Qian, Privacy preserving machine learning with homomorphic encryption and federated learning, Futur. Internet 13 (4) (2021) 94.

[2] J. Ma, S.-A. Naas, S. Sigg, X. Lyu, Privacy-preserving federated learning based on multi-key homomorphic encryption, Int. J. Intell. Syst. 37 (9) (2022) 5880–5901.

[3] J.O. Gidiagba, N.K. Nwaobia, P.W. Biu, C.A. Ezeigweneme, A.A. Umoh, Review on the evolution and impact of iot-driven predictive maintenance: assessing advancements, their role in enhancing system longevity, and sustainable operations in both mechanical and electrical realms, Comput. Sci. & IT Res. J. 5 (1) (2024) 166–189.

[4] J. Park, H. Lim, Privacy-preserving federated learning using homomorphic encryption, Appl. Sci. 12 (2) (2022) 734.

[5] L. Zhang, J. Xu, P. Vijayakumar, P.K. Sharma, U. Ghosh, Homomorphic encryption-based privacy-preserving federated learning in IoT-enabled healthcare system, IEEE Trans. Netw. Sci. Eng. 10 (5) (2022) 2864–2880.

[6] B. Jia, X. Zhang, J. Liu, Y. Zhang, K. Huang, Y. Liang, Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT, IEEE Trans. Ind. Inform. 18 (6) (2021) 4049–4058.

[7] H. Ran, X. Gao, L. Li, W. Li, S. Tian, G. Wang, H. Shi, X. Ning, Brain-inspired fast-and slow-update prompt tuning for few-shot class-incremental learning, IEEE Trans. Neural Netw. Learn. Syst. (2024).

[8] F. Wibawa, F.O. Catak, M. Kuzlu, S. Sarp, U. Cali, Homomorphic encryption and federated learning based privacy-preserving cnn training: Covid-19 detection use-case, in: Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference, 2022, pp. 85–90.

[9] H. Ku, W. Susilo, Y. Zhang, W. Liu, M. Zhang, Privacy-preserving federated learning in medical diagnosis with homomorphic re-encryption, Comput. Stand. Interfaces 80 (2022) 103583.

[10] Y. Bai, M. Fan, A method to improve the privacy and security for federated learning, in: 2021 IEEE 6th International Conference on Computer and Communication Systems, ICCCS, IEEE, 2021, pp. 704–708.

[11] X. Liu, H. Li, G. Xu, R. Lu, M. He, Adaptive privacy-preserving federated learning, Peer-To-Peer Netw. Appl. 13 (2020) 2356–2366.

[12] W. Jin, Y. Yao, S. Han, C. Joe-Wong, S. Ravi, S. Avestimehr, C. He, FedML-HE: An efficient homomorphic-encryption-based privacy-preserving federated learning system, 2023, arXiv preprint arXiv:2303.10837.

[13] G. Nzeako, C.D. Okeke, M.O. Akinsanya, O.A. Popoola, E.G. Chukwurah, Security paradigms for IoT in telecom networks: Conceptual challenges and solution pathways, Eng. Sci. Technol. J. 5 (5) (2024) 1606–1626.

[14] C. Li, J. Wang, S. Wang, Y. Zhang, A review of IoT applications in healthcare, Neurocomputing 565 (2024) 127017.

[15] X. Zhang, A. Fu, H. Wang, C. Zhou, Z. Chen, A privacy-preserving and verifiable federated learning scheme, in: ICC 2020-2020 IEEE International Conference on Communications, ICC, IEEE, 2020, pp. 1–6.

[16] M. Hao, H. Li, G. Xu, S. Liu, H. Yang, Towards efficient and privacy-preserving federated deep learning, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–6.

[17] Q. Chen, F. He, G. Wang, X. Bai, L. Cheng, X. Ning, Dual guidance enabled fuzzy inference for enhanced fine-grained recognition, IEEE Trans. Fuzzy Syst. (2024) 1–14, http://dx.doi.org/10.1109/TFUZZ.2024.3427654.

[18] N.U. Prince, M.A. Al Mamun, A.O. Olajide, O.U. Khan, A.B. Akeem, A.I. Sani, IEEE standards and deep learning techniques for securing internet of things (IoT) devices against cyber attacks, J. Comput. Anal. Appl. 33 (7) (2024).

[19] X. Wang, Z. Wan, A. Hekmati, M. Zong, S. Alam, M. Zhang, B. Krishnamachari, Iot in the era of generative ai: Vision and challenges, 2024, arXiv preprint arXiv:2401.01923.

[20] M.O. Akinsanya, C.C. Ekechi, C.D. Okeke, Security paradigms for iot in telecom networks: conceptual challenges and solution pathways, Eng. Sci. Technol. J. 5 (4) (2024) 1431–1451.

[21] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, B. He, A survey on federated learning systems: Vision, hype and reality for data privacy and protection, IEEE Trans. Knowl. Data Eng. 35 (4) (2021) 3347–3366.

[22] L. Ge, H. Li, X. Wang, Z. Wang, A review of secure federated learning: privacy leakage threats, protection technologies, challenges and future directions, Neurocomputing (2023) 126897.

[23] Z. Li, V. Sharma, S.P. Mohanty, Preserving data privacy via federated learning: Challenges and solutions, IEEE Consumer Electron. Mag. 9 (3) (2020) 8–16.

[24] R. Wang, J. Lai, Z. Zhang, X. Li, P. Vijayakumar, M. Karuppiah, Privacy-preserving federated learning for internet of medical things under edge computing, IEEE J. Biomed. Health Inf. 27 (2) (2022) 854–865.

[25] S. Bharati, M. Mondal, P. Podder, V. Prasath, Federated learning: Applications, challenges and future directions, Int. J. Hybrid Intell. Syst. 18 (1–2) (2022) 19–35.

[26] Y. Chen, J. Li, F. Wang, K. Yue, Y. Li, B. Xing, L. Zhang, L. Chen, DS2PM: A data-sharing privacy protection model based on blockchain and federated learning, IEEE Internet Things J. 10 (14) (2021) 12112–12125.

[27] R. Yang, T. Zhao, F.R. Yu, M. Li, D. Zhang, X. Zhao, Blockchain-based federated learning with enhanced privacy and security using homomorphic encryption and reputation, IEEE Internet Things J. (2024).

[28] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, B. Thorne, Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption, 2017, arXiv preprint arXiv:1711.10677.

[29] L. Yin, J. Feng, H. Xun, Z. Sun, X. Cheng, A privacy-preserving federated learning for multiparty data sharing in social IoTs, IEEE Trans. Netw. Sci. Eng. 8 (3) (2021) 2706–2718.

[30] M. Asad, A. Moustafa, T. Ito, Fedopt: Towards communication efficiency and privacy preservation in federated learning, Appl. Sci. 10 (8) (2020) 2864.

[31] J. Zhang, B. Chen, S. Yu, H. Deng, PEFL: A privacy-enhanced federated learning scheme for big data analytics, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6.

[32] H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, H. Möllering, T.D. Nguyen, P. Rieger, A.-R. Sadeghi, T. Schneider, H. Yalame, et al., SAFELearn: Secure aggregation for private federated learning, in: 2021 IEEE Security and Privacy Workshops, SPW, IEEE, 2021, pp. 56–62.

[33] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, Y. Zhou, A hybrid approach to privacy-preserving federated learning, in: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, 2019, pp. 1–11.

[34] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, S. Liu, Efficient and privacy-enhanced federated learning for industrial artificial intelligence, IEEE Trans. Ind. Inform. 16 (10) (2019) 6532–6542.

[35] X. Gu, F. Sabrina, Z. Fan, S. Sohail, A review of privacy enhancement methods for federated learning in healthcare systems, Int. J. Environ. Res. Public Health 20 (15) (2023) 6539.

[36] Y. Li, X. Tao, X. Zhang, J. Liu, J. Xu, Privacy-preserved federated learning for autonomous driving, IEEE Trans. Intell. Transp. Syst. 23 (7) (2021) 8423–8434.

[37] L. Li, Y. Fan, M. Tse, K.-Y. Lin, A review of applications in federated learning, Comput. Ind. Eng. 149 (2020) 106854.

[38] Z. Zhang, L. Zhang, Q. Li, K. Wang, N. He, T. Gao, Privacy-enhanced momentum federated learning via differential privacy and chaotic system in industrial cyber–physical systems, ISA Trans. 128 (2022) 17–31.

[39] Y. Li, Y. Zhou, A. Jolfaei, D. Yu, G. Xu, X. Zheng, Privacy-preserving federated learning framework based on chained secure multiparty computing, IEEE Internet Things J. 8 (8) (2020) 6178–6186.

[40] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, W. Zhang, A survey on federated learning: challenges and applications, Int. J. Mach. Learn. Cybern. 14 (2) (2023) 513–535.

[41] B. Zhao, K. Fan, K. Yang, Z. Wang, H. Li, Y. Yang, Anonymous and privacy-preserving federated learning with industrial big data, IEEE Trans. Ind. Inform. 17 (9) (2021) 6314–6323.

[42] Y. LeCun, The MNIST database of handwritten digits, 1998, http://yann.lecun.com/exdb/mnist/.