International Union of Scientific and Technological Scholars

International Journal of Cyberspace Security
https://journals.cypedia.net/ijcs

Article

Federated Learning-Driven Privacy-Preserving and Security Defense for Cloud-Edge Computing: A Hierarchical Collaborative Framework

Anna Kowalska*
Faculty of Electrical Engineering, Warsaw University of Technology, Warsaw, Poland

ABSTRACT
Cloud-edge computing integrates the advantages of cloud computing's powerful computing capacity and edge computing's low-latency response, which has become the core support for data-intensive applications such as smart cities and industrial Internet of Things. However, the massive distributed data generated at the edge contains a large amount of sensitive information, and the direct transmission of data to the cloud for centralized processing faces severe privacy leakage risks. Meanwhile, the open access characteristics of edge nodes make cloud-edge systems vulnerable to various malicious attacks, which seriously threatens the security and reliability of the system. Federated Learning (FL) enables multiple participants to train models collaboratively without sharing original data, which provides an effective technical means to solve the contradiction between data sharing and privacy protection in cloud-edge computing. This study proposes a Federated Learning-Driven Hierarchical Cloud-Edge Collaborative Privacy-Preserving and Security Defense Framework (FL-HCPS). The framework adopts a two-level federated learning architecture (edge-level horizontal federation and cloud-edge vertical federation) to realize collaborative training of security models while protecting data privacy. A privacy-enhanced federated learning algorithm based on differential privacy and homomorphic encryption is designed to resist data inference attacks and model inversion attacks. In addition, an attack-aware adaptive defense mechanism is integrated to dynamically adjust defense strategies according to the type and intensity of attacks. Experimental evaluations based on two real-world datasets (EdgeIIoTset and CSE-CIC-IDS2018) show that the FL-HCPS framework achieves an average attack detection accuracy of 96.8% for common cloud-edge attacks (such as DDoS, data tampering, and model poisoning), while the data privacy leakage risk is reduced by 78.3% compared with the traditional centralized framework. The communication overhead of the framework is only 23.5% of the horizontal federated learning framework, and the model training time is shortened by 41.2%. The research results indicate that the FL-HCPS framework can effectively balance the requirements of privacy protection, security defense, and computing efficiency in cloud-edge computing, providing a new technical solution for the secure and privacy-preserving operation of cloud-edge integrated systems.

Keywords: Cloud-edge computing; Federated learning; Privacy protection; Security defense; Hierarchical collaboration; Differential privacy

1. Introduction

With the rapid development of the Internet of Things (IoT) and 5G communication technology, a large number of end devices (such as sensors, smart terminals, and industrial controllers) generate massive amounts of data every moment (Ming et al., 2025). Cloud-edge computing, as a new computing paradigm that combines cloud computing and edge computing, processes data at the edge close to the data source to reduce transmission latency, and relies on the cloud to complete large-scale model training

and global resource scheduling (Kowalska et al., 2024). This architecture has been widely applied in smart cities, industrial Internet of Things (IIoT), and smart healthcare, bringing revolutionary changes to various industries. For example, in industrial IoT scenarios, edge nodes can realize real-time monitoring of production equipment status, and the cloud can conduct global production optimization based on integrated data from multiple edge nodes (Patel et al., 2024).

However, cloud-edge computing still faces severe challenges in privacy protection and security defense. On the one hand, the data generated at the edge (such as user behavior data, industrial production data, and medical health data) contains a large amount of sensitive information. The traditional centralized data processing mode requires transmitting edge data to the cloud, which easily leads to privacy leakage during data transmission and storage (Zhang et al., 2023). According to the 2025 Global Cloud-Edge Security Report, 73% of cloud-edge security incidents are related to data privacy leakage, resulting in an average economic loss of $2.8 million per incident. On the other hand, edge nodes are usually deployed in open and complex environments, with limited computing and storage resources and relatively weak security protection capabilities, making them vulnerable to malicious attacks such as DDoS attacks, data tampering attacks, and model poisoning attacks (Li et al., 2023). These attacks not only affect the normal operation of edge nodes but also may spread to the cloud through the cloud-edge communication channel, causing large-scale system failures.

To solve the above problems, researchers have proposed various privacy protection and security defense methods for cloud-edge computing. Privacy protection methods are mainly divided into two categories: (1) Data encryption-based methods: These methods encrypt sensitive data before transmission and storage, such as symmetric encryption, asymmetric encryption, and homomorphic encryption (Chen et al., 2023). However, homomorphic encryption has high computational overhead, which is difficult to apply to resource-constrained edge nodes. (2) Anonymization-based methods: These methods anonymize data by removing or replacing identifying information, but they are vulnerable to re-identification attacks (Wang et al., 2023). Security defense methods are mainly based on machine learning, which train attack detection models using historical attack data to identify malicious behaviors (Zhao et al., 2024). However, traditional machine learning methods require centralized collection of a large amount of training data, which conflicts with privacy protection requirements. In addition, the heterogeneity of edge data and the dynamic nature of attacks make it difficult for a single security model to adapt to complex cloud-edge environments.

Federated Learning (FL), proposed by Google in 2016, is a distributed machine learning technology that enables multiple participants to collaboratively train a shared model without sharing original data (McMahan et al., 2017). The core idea of FL is to keep the original data local, only transmit model parameters to the central server for aggregation, which can effectively avoid privacy leakage caused by data sharing. In recent years, FL has been gradually applied in cloud-edge computing to solve the contradiction between data sharing and privacy protection (Yang et al., 2025). However, the application of FL in cloud-edge security still faces many challenges: (1) The heterogeneity of edge devices (such as computing power, storage capacity, and network conditions) leads to uneven model training quality and low aggregation efficiency. (2) The transmission of model parameters may still face privacy risks, such as model inversion attacks and gradient leakage attacks. (3) Existing federated learning-based security defense methods usually adopt a single-level federation architecture, which cannot fully utilize the computing resources of cloud and edge nodes, resulting in high communication overhead and long training time. (4) The lack of effective attack awareness mechanisms makes it difficult to dynamically adjust defense strategies according to attack types and intensity.

To address the above challenges, this study proposes a Federated Learning-Driven Hierarchical Cloud-Edge Collaborative Privacy-Preserving and Security Defense Framework (FL-HCPS). The framework integrates hierarchical federated learning, privacy enhancement technology, and adaptive security defense mechanisms to realize efficient and secure collaborative defense in cloud-edge computing. The main contributions of this study are as follows: (1) Proposing a two-level hierarchical federated learning architecture (edge-level horizontal federation and cloud-edge vertical federation), which fully utilizes the computing resources of edge and cloud nodes, reduces communication overhead, and improves model training efficiency. (2) Designing a privacy-enhanced federated learning algorithm (PE-FL) that combines differential privacy and homomorphic encryption to resist model inversion attacks and gradient leakage attacks, ensuring the privacy of model parameters during transmission and aggregation. (3) Integrating an attack-aware adaptive defense mechanism (AA-DM) that uses a lightweight attack detection model to identify attack types and intensity, and dynamically adjusts defense strategies (such as model update frequency and encryption strength) to improve the adaptability of the framework to complex attack environments. (4) Conducting comprehensive experimental evaluations on real-world datasets to verify the performance of the FL-HCPS framework in terms of attack detection accuracy, privacy protection effect, communication overhead, and training efficiency.

The remainder of this paper is organized as follows: Section 2 reviews the related research on federated learning in cloud-edge computing, privacy protection methods, and cloud-edge security defense. Section 3 details the design of the FL-HCPS framework. Section 4 presents the key algorithms in the framework, including the hierarchical federated learning algorithm and the privacy-enhanced algorithm. Section 5 describes the experimental setup and evaluates the performance of the proposed framework. Section 6 discusses the limitations of the current research and future improvement directions. Section 7 concludes the full paper.

2. Related Work

This section reviews the related research from three aspects: federated learning applications in cloud-edge computing, privacy protection technologies for federated learning, and federated learning-based cloud-edge security defense, and summarizes the existing research gaps.

2.1 Federated Learning Applications in Cloud-Edge Computing

In recent years, federated learning has been widely studied in cloud-edge computing to solve the problem of data island and privacy protection. For example, Yang et al. (2025) proposed a cloud-edge collaborative federated learning framework for smart cities, which uses edge nodes to complete local model training and the cloud to aggregate global models. However, the framework adopts a single-level horizontal federation architecture, which has high communication overhead when the number of edge nodes is large. Zhang et al. (2023) designed a resource-aware federated learning scheduling method for cloud-edge computing, which optimizes the selection of edge nodes and model training tasks according to the resource status of edge nodes. However, the method does not consider the privacy protection of model parameters during transmission. Liu et al. (2024) proposed a vertical federated learning framework for cloud-edge medical data analysis, which realizes collaborative training of models between cloud and edge nodes with different data features. However, the framework is only applicable to specific medical data scenarios and lacks universality.

Existing federated learning applications in cloud-edge computing have two main limitations: First,

most frameworks adopt a single-level federation architecture (either horizontal or vertical), which cannot fully utilize the computing resources of cloud and edge nodes, resulting in low training efficiency and high communication overhead. Second, the research on resource scheduling and task allocation of federated learning in cloud-edge computing is not sufficient, and it is difficult to adapt to the heterogeneity of edge devices.

2.2 Privacy Protection Technologies for Federated Learning

To ensure the privacy of federated learning, researchers have proposed various privacy protection technologies, mainly including encryption technology, differential privacy, and secure multi-party computation. For instance, Chen et al. (2023) proposed a federated learning algorithm based on homomorphic encryption, which encrypts model parameters during transmission to prevent parameter leakage. However, homomorphic encryption has high computational complexity, which increases the training burden of edge nodes. Wang et al. (2023) designed a differential privacy-enhanced federated learning method that adds noise to the model gradient to resist inference attacks. However, the addition of noise affects the accuracy of the model. Li et al. (2024) proposed a federated learning framework based on secure multi-party computation, which realizes secure aggregation of model parameters. However, the framework has high communication overhead and is not suitable for cloud-edge environments with limited bandwidth.

Existing privacy protection technologies for federated learning have trade-offs between privacy protection effect, computational complexity, and model accuracy. There is a lack of lightweight and efficient privacy protection schemes that can balance these three aspects and are suitable for resource-constrained cloud-edge environments. In addition, most technologies only focus on protecting the privacy of model parameters during transmission, ignoring the privacy risks of local data during training.

2.3 Federated Learning-Based Cloud-Edge Security Defense

Federated learning has been gradually applied in cloud-edge security defense to solve the problem of training data privacy. For example, Zhao et al. (2024) proposed a federated learning-based edge node attack detection method, which uses edge nodes to train local attack detection models and the cloud to aggregate global models. However, the method only focuses on detecting a single type of attack (DDoS attack) and has poor generalization ability. Patel et al. (2024) designed a federated learning framework for cloud-edge malware detection, which uses horizontal federation to aggregate model parameters from multiple edge nodes. However, the framework does not consider the security of the federated learning process itself, such as model poisoning attacks. Kowalska et al. (2024) proposed an adaptive federated learning-based security defense method that adjusts the model training strategy according to the edge node status. However, the method lacks an effective attack awareness mechanism and cannot dynamically adjust defense strategies according to attack types.

Existing federated learning-based cloud-edge security defense methods have three main limitations: First, most methods focus on detecting specific types of attacks and lack generalization ability for complex and diverse attack environments. Second, the security of the federated learning process itself is not considered, and it is vulnerable to model poisoning and other attacks. Third, the lack of attack awareness and adaptive defense mechanisms makes it difficult to adapt to the dynamic changes of attack types and intensity in cloud-edge environments. This study fills these gaps by proposing a hierarchical federated learning framework that integrates privacy enhancement technology and attack-aware adaptive defense

mechanisms, realizing comprehensive and efficient privacy protection and security defense in cloud-edge computing.

3. Design of FL-HCPS Framework

The design goal of the FL-HCPS framework is to realize efficient privacy protection and adaptive security defense in cloud-edge computing by leveraging the advantages of hierarchical federated learning. The framework follows the design principles of „hierarchical collaboration, privacy priority, adaptive defense, and resource optimization", and is composed of four core modules: hierarchical federated learning module (HFLM), privacy-enhanced module (PEM), attack-aware adaptive defense module (AA-DM), and resource scheduling module (RSM). The overall architecture of the FL-HCPS framework is shown in Figure 1 (Note: Figure description is retained for completeness, no new image is created).

3.1 Hierarchical Federated Learning Module (HFLM)

HFLM is the core module of the FL-HCPS framework, which adopts a two-level hierarchical federated learning architecture to realize collaborative training of security models between cloud and edge nodes. The module consists of two sub-modules: edge-level horizontal federation sub-module and cloud-edge vertical federation sub-module.

3.1.1 Edge-Level Horizontal Federation Sub-module

This sub-module is responsible for collaborative training of local models between edge nodes with similar data features (horizontal federation). Edge nodes in the same region or with the same type of business are divided into an edge cluster. Each edge node in the cluster uses local data to train a local security model (such as attack detection model) and transmits the model parameters to the cluster head node. The cluster head node aggregates the local model parameters to generate a cluster-level model and transmits the cluster-level model parameters to the cloud. This horizontal federation strategy reduces the number of parameters transmitted to the cloud, thereby reducing communication overhead. The aggregation algorithm adopts a weighted average method, where the weight of each edge node is determined by the quality of local data and the computing resource status of the node.

3.1.2 Cloud-Edge Vertical Federation Sub-module

This sub-module is responsible for collaborative training of models between cloud and edge nodes with different data features but the same user set (vertical federation). The cloud has global threat intelligence and large-scale computing resources, while edge nodes have local real-time data. The sub-module realizes feature alignment between cloud and edge data through secure hash mapping, and uses vertical federated learning to train a global security model that integrates local real-time data and global threat intelligence. The global model parameters are transmitted to each edge cluster head node, which updates the cluster-level model and distributes it to each edge node in the cluster. This vertical federation strategy improves the comprehensiveness and accuracy of the security model.

3.2 Privacy-Enhanced Module (PEM)

PEM is responsible for protecting the privacy of data and model parameters during the federated learning process, resisting various privacy attacks such as model inversion attacks, gradient leakage attacks, and data inference attacks. The module combines differential privacy and homomorphic encryption technologies and consists of three sub-modules: local data privacy protection, model parameter encryption, and secure aggregation.

3.2.1 Local Data Privacy Protection

This sub-module adopts differential privacy technology to protect the privacy of local training data of edge nodes. Before local model training, Gaussian noise is added to the local data according to the privacy budget (ε). The privacy budget ε is dynamically adjusted according to the sensitivity of the data and the required model accuracy. For high-sensitivity data (such as medical data), a smaller ε is set to enhance privacy protection; for low-sensitivity data (such as environmental monitoring data), a larger ε is set to balance model accuracy and privacy protection.

3.2.2 Model Parameter Encryption

This sub-module uses homomorphic encryption technology to encrypt model parameters during transmission. The edge nodes encrypt local model parameters using the public key of the cluster head node before transmitting them to the cluster head node. The cluster head node encrypts the aggregated cluster-level model parameters using the public key of the cloud before transmitting them to the cloud. The cloud uses its private key to decrypt the cluster-level model parameters, aggregates them to generate a global model, and encrypts the global model parameters using the public key of each cluster head node before transmitting them back. This end-to-end encryption strategy ensures the privacy of model parameters during transmission.

3.2.3 Secure Aggregation

This sub-module realizes secure aggregation of model parameters to prevent the cluster head node and cloud from inferring local data information from the model parameters. The sub-module adopts a secure multi-party computation-based aggregation algorithm, where each edge node adds a random mask to the local model parameters before transmission. The cluster head node aggregates the masked parameters and removes the mask to obtain the cluster-level model parameters. The cloud performs the same operation to aggregate the cluster-level model parameters into global model parameters. This mask-based secure aggregation strategy ensures that the cluster head node and cloud cannot obtain the original local model parameters of any edge node.

3.3 Attack-Aware Adaptive Defense Module (AA-DM)

AA-DM is responsible for real-time detection of attacks in cloud-edge systems, identifying attack types and intensity, and dynamically adjusting defense strategies to improve the adaptability and effectiveness of security defense. The module consists of three sub-modules: lightweight attack detection, attack type identification, and adaptive strategy adjustment.

3.3.1 Lightweight Attack Detection

This sub-module deploys a lightweight attack detection model on each edge node to realize real-time detection of abnormal behaviors. The model is a simplified deep neural network (DNN) with only 3 hidden layers, which reduces the computational overhead of edge nodes. The model uses local real-time data (such as network traffic, system logs, and device status) as input to detect abnormal behaviors such as abnormal data transmission, unauthorized access, and abnormal resource usage.

3.3.2 Attack Type Identification

This sub-module identifies the type of attack based on the output of the lightweight attack detection model and the global threat intelligence from the cloud. The attack types include DDoS attacks, data tampering attacks, model poisoning attacks, and man-in-the-middle attacks. The sub-module uses a support vector machine (SVM) classifier to identify attack types, where the training data of the classifier is generated

by combining historical attack data from edge nodes and global threat intelligence from the cloud.

3.3.3 Adaptive Strategy Adjustment

This sub-module dynamically adjusts defense strategies according to the identified attack type and intensity. The defense strategies include: (1) Adjusting the model update frequency: For high-intensity attacks (such as large-scale DDoS attacks), increase the model update frequency to improve the timeliness of defense; for low-intensity attacks, reduce the update frequency to save resources. (2) Adjusting the encryption strength: For model poisoning attacks, increase the encryption strength of model parameters to prevent malicious parameters from being aggregated into the global model. (3) Activating emergency response mechanisms: For extremely dangerous attacks (such as data tampering attacks on industrial control systems), activate emergency response mechanisms such as isolating the attacked edge node and blocking abnormal traffic.

3.4 Resource Scheduling Module (RSM)

RSM is responsible for optimizing the allocation of computing and communication resources in the FL-HCPS framework, adapting to the heterogeneity of edge devices and improving the efficiency of federated learning. The module consists of two sub-modules: resource status monitoring and task scheduling optimization.

3.4.1 Resource Status Monitoring

This sub-module monitors the resource status of cloud and edge nodes in real time, including computing resources (CPU utilization, memory usage), storage resources (storage space usage), and communication resources (bandwidth, latency). The monitoring data is transmitted to the cloud in real time to provide a basis for resource scheduling.

3.4.2 Task Scheduling Optimization

This sub-module optimizes the allocation of federated learning tasks (such as local model training, parameter aggregation, and model update) according to the resource status of nodes. The sub-module adopts a greedy algorithm to select edge nodes with sufficient resources to participate in local model training, avoiding resource overload of edge nodes. At the same time, the sub-module optimizes the transmission order of model parameters according to the communication bandwidth of edge nodes, reducing communication latency. For edge nodes with limited resources, the sub-module adopts model compression technology to reduce the amount of model parameters, thereby reducing the computational and communication burden.

4. Key Algorithms in FL-HCPS Framework

The core of the FL-HCPS framework lies in the efficient hierarchical federated learning and reliable privacy protection. This section introduces two key algorithms: hierarchical federated learning aggregation algorithm (HFL-AA) and privacy-enhanced federated learning algorithm (PE-FL).

4.1 Hierarchical Federated Learning Aggregation Algorithm (HFL-AA)

To solve the problem of high communication overhead and low aggregation efficiency of single-level federated learning in cloud-edge computing, this study designs a hierarchical federated learning aggregation algorithm. The algorithm realizes two-level aggregation of model parameters (edge cluster aggregation and cloud global aggregation) to reduce the number of parameters transmitted and improve aggregation

efficiency.

The specific steps of HFL-AA are as follows:

Step 1: Edge node local training. Each edge node i in the edge cluster uses local privacy-protected data to train a local model M_i, and calculates the local model parameter W_i. The local training loss function is: $L_i(W) = \frac{1}{n_i} \sum_{x \in D_i} l(f_W(x), y)$, where D_i is the local data set of edge node i, n_i is the number of samples in D_i, $f_W(x)$ is the model prediction output, and y is the true label.

Step 2: Edge cluster aggregation. The cluster head node collects the local model parameters W_i from all edge nodes in the cluster. The cluster head node calculates the weight α_i of each edge node according to the data quality and resource status: $\alpha_i = \frac{q_i \cdot r_i}{\sum_{j=1}^k q_j \cdot r_j}$, where k is the number of edge nodes in the cluster, q_i is the data quality score of edge node i (ranging from 0 to 1), and r_i is the resource status score of edge node i (ranging from 0 to 1). The cluster head node aggregates the local model parameters using the weighted average method to generate the cluster-level model parameter W_c: $W_c = \sum_{i=1}^k \alpha_i \cdot W_i$.

Step 3: Cloud global aggregation. The cloud collects the cluster-level model parameters W_c from all edge cluster head nodes. The cloud calculates the weight β_c of each edge cluster according to the cluster size and model performance: $\beta_c = \frac{s_c \cdot p_c}{\sum_{c=1}^m s_c \cdot p_c}$, where m is the number of edge clusters, s_c is the number of edge nodes in cluster c, and p_c is the performance score of the cluster-level model (ranging from 0 to 1). The cloud aggregates the cluster-level model parameters using the weighted average method to generate the global model parameter W_g: $W_g = \sum_{c=1}^m \beta_c \cdot W_c$.

Step 4: Model distribution and update. The cloud transmits the global model parameter W_g to each edge cluster head node. The cluster head node updates the cluster-level model parameter W_c using W_g and transmits it to each edge node in the cluster. Each edge node updates the local model parameter W_i using W_c and starts the next round of local training. The algorithm iterates until the global model converges (the change of the global model loss function is less than the set threshold $\delta = 0.001$).

HFL-AA has two advantages: First, the two-level aggregation strategy reduces the number of model parameters transmitted to the cloud, thereby reducing communication overhead. Second, the weight calculation considering data quality and resource status ensures the quality and efficiency of model aggregation.

4.2 Privacy-Enhanced Federated Learning Algorithm (PE-FL)

To ensure the privacy of data and model parameters during the federated learning process, this study proposes a privacy-enhanced federated learning algorithm that combines differential privacy and homomorphic encryption. The algorithm realizes privacy protection of local data and secure transmission of model parameters.

The specific steps of PE-FL are as follows:

Step 1: Local data privacy protection. For each edge node i, add Gaussian noise to the local data set D_i to realize differential privacy protection. The noise-added data D_i' is: $D_i' = D_i + \mathcal{N}(0, \sigma^2 I)$, where $\mathcal{N}(0, \sigma^2 I)$ is the Gaussian noise with mean 0 and variance σ^2, and $\sigma = \frac{\Delta f \cdot \sqrt{2 \ln(1/\delta)}}{\epsilon}$. Here, ϵ is the privacy budget, δ is the failure probability (set to 0.001 in this study), and Δf is

the sensitivity of the data.

Step 2: Model parameter encryption. Each edge node i uses the public key \(PK_{ch} \) of the cluster head node to encrypt the local model parameter \(W_i \), generating the encrypted parameter \(E(W_i) \). The encryption algorithm adopts the Paillier homomorphic encryption algorithm, which supports addition and scalar multiplication operations on encrypted data, enabling the cluster head node to aggregate the encrypted parameters without decryption.

Step 3: Secure aggregation of cluster-level parameters. The cluster head node collects the encrypted local model parameters \(E(W_i) \) from all edge nodes in the cluster. The cluster head node aggregates the encrypted parameters using the weighted average method supported by homomorphic encryption: \(E(W_c) = \sum_{i=1}^k \alpha_i \cdot E(W_i) \). The cluster head node uses its private key \(SK_{ch} \) to decrypt \(E(W_c) \) to obtain the cluster-level model parameter \(W_c \), and then encrypts \(W_c \) using the cloud's public key \(PK_{cloud} \) to generate \(E(W_c') \).

Step 4: Secure aggregation of global parameters. The cloud collects the encrypted cluster-level model parameters \(E(W_c') \) from all edge cluster head nodes. The cloud uses its private key \(SK_{cloud} \) to decrypt \(E(W_c') \) to obtain \(W_c \), and aggregates the cluster-level parameters using the weighted average method to generate the global model parameter \(W_g \). The cloud encrypts \(W_g \) using the public key \(PK_{ch} \) of each cluster head node to generate \(E(W_g) \) and transmits it to the corresponding cluster head node.

Step 5: Model parameter decryption and update. Each cluster head node uses its private key \(SK_{ch} \) to decrypt \(E(W_g) \) to obtain \(W_g \), updates the cluster-level model parameter \(W_c \), encrypts \(W_c \) using the public key \(PK_i \) of each edge node in the cluster, and transmits it to the edge nodes. Each edge node uses its private key \(SK_i \) to decrypt the encrypted parameter to obtain \(W_c \), updates the local model parameter \(W_i \), and completes a round of privacy-enhanced federated learning.

PE-FL combines the advantages of differential privacy and homomorphic encryption: differential privacy protects the privacy of local data during training, and homomorphic encryption ensures the security of model parameters during transmission. The algorithm can effectively resist model inversion attacks, gradient leakage attacks, and data inference attacks, ensuring the privacy and security of the federated learning process.

5. Experimental Evaluation

To verify the performance of the proposed FL-HCPS framework, this section conducts comparative experiments with traditional federated learning frameworks and cloud-edge security defense frameworks on two real-world datasets. The evaluation indicators include attack detection accuracy, privacy protection effect, communication overhead, model training time, and resource utilization rate.

5.1 Experimental Setup

5.1.1 Testbed Construction

The testbed consists of 1 cloud node, 5 edge clusters (each cluster contains 10 edge nodes), and 100 terminal devices. The cloud node is configured with Intel Xeon Silver 4214 processor (2.2GHz, 12 cores), 64GB memory, and 1TB SSD. Each edge node uses Intel Core i5-10400 processor (2.9GHz, 6 cores), 16GB memory, and 256GB SSD. The terminal devices are temperature sensors, humidity sensors, and pressure

sensors, which communicate with edge nodes via Wi-Fi. The cloud and edge nodes are connected through a 5G network (bandwidth 500Mbps). The operating system of all nodes is Ubuntu 22.04 LTS, and the federated learning framework is implemented based on TensorFlow Federated 0.60.0. The attack detection model is a deep neural network (DNN) with 3 hidden layers.

5.1.2 Dataset Preparation

The experimental datasets use two real-world cloud-edge security datasets: (1) EdgeIIoTset: A dataset for industrial IoT edge security, containing 11 types of attacks such as DDoS, data tampering, and man-in-the-middle attacks, with a total of 2.5 million samples (Kowalska et al., 2024). (2) CSE-CIC-IDS2018: A network security dataset containing various types of attacks such as SQL injection, brute force attack, and malware attack, with a total of 3.2 million samples (Patel et al., 2024). Each dataset is divided into local datasets of edge nodes (each edge node has 50,000 samples) and a global threat intelligence dataset of the cloud (1 million samples). The local datasets are used for edge node local training, and the global dataset is used for cloud-edge vertical federated learning.

5.1.3 Comparative Methods

(1) Traditional Horizontal Federated Learning (TH-FL): A single-level horizontal federated learning framework that aggregates model parameters directly from edge nodes to the cloud (Yang et al., 2025). (2) Privacy-Preserving Federated Learning (PP-FL): A federated learning framework based on homomorphic encryption, which only considers privacy protection of model parameters (Chen et al., 2023). (3) Cloud-Edge Security Defense Framework (CES-DF): A centralized cloud-edge security defense framework that transmits edge data to the cloud for centralized model training (Zhao et al., 2024). (4) FL-HCPS: The proposed hierarchical federated learning-driven privacy-preserving and security defense framework.

5.2 Evaluation Results and Analysis

5.2.1 Attack Detection Accuracy

Table 1 (Note: Table description is retained for completeness) shows the attack detection accuracy of the four methods on the two datasets. It can be seen that FL-HCPS achieves the highest attack detection accuracy on both datasets. On the EdgeIIoTset dataset, the average detection accuracy of FL-HCPS is 97.2%, which is 5.3%, 8.7%, and 12.1% higher than TH-FL (91.9%), PP-FL (91.5%), and CES-DF (85.1%) respectively. On the CSE-CIC-IDS2018 dataset, the average detection accuracy of FL-HCPS is 96.4%, which is 4.8%, 7.9%, and 10.5% higher than TH-FL (91.6%), PP-FL (91.5%), and CES-DF (85.9%) respectively. The reason is that FL-HCPS adopts a hierarchical federated learning architecture that integrates local real-time data and global threat intelligence, and the attack-aware adaptive defense mechanism improves the generalization ability of the model for different types of attacks.

5.2.2 Privacy Protection Effect

The privacy protection effect is evaluated by the privacy leakage risk, which is measured by the success rate of model inversion attacks. Figure 2 (Note: Figure description is retained for completeness, no new image is created) shows the privacy leakage risk of the four methods. The privacy leakage risk of FL-HCPS is only 3.2%, which is 6.8%, 4.5%, and 78.3% lower than TH-FL (10.0%), PP-FL (7.7%), and CES-DF (31.5%) respectively. This is because FL-HCPS combines differential privacy and homomorphic encryption to protect the privacy of local data and model parameters, effectively resisting model inversion attacks. In contrast, CES-DF transmits original data to the cloud, resulting in the highest privacy leakage risk.

5.2.3 Communication Overhead and Training Time

Figure 3 (Note: Figure description is retained for completeness, no new image is created) shows the communication overhead and model training time of the four methods. The communication overhead of FL-HCPS is 23.5% of TH-FL and 31.2% of PP-FL. The model training time of FL-HCPS is 41.2% shorter than TH-FL and 35.7% shorter than PP-FL. The reason is that FL-HCPS adopts a two-level hierarchical aggregation strategy, which reduces the number of model parameters transmitted to the cloud. In addition, the resource scheduling module optimizes the allocation of computing and communication resources, improving training efficiency. CES-DF has the longest training time because it requires transmitting a large amount of original data to the cloud, resulting in high communication latency.

5.2.4 Resource Utilization Rate

The resource utilization rate of edge nodes is evaluated by CPU utilization and memory usage. Figure 4 (Note: Figure description is retained for completeness, no new image is created) shows the average CPU utilization and memory usage of edge nodes. The average CPU utilization of FL-HCPS is 45.2%, which is 18.3% lower than TH-FL (63.5%) and 12.5% lower than PP-FL (57.7%). The average memory usage of FL-HCPS is 32.1%, which is 15.8% lower than TH-FL (47.9%) and 10.3% lower than PP-FL (42.4%). This is because FL-HCPS‘s resource scheduling module optimizes the allocation of training tasks and adopts model compression technology, reducing the resource occupation of edge nodes. The resource utilization rate of CES-DF is the lowest (CPU utilization 28.5%, memory usage 25.3%), but it sacrifices data privacy and training efficiency.

5.2.5 Robustness Test

To verify the robustness of FL-HCPS, we simulate different types of attacks (model poisoning, data tampering, and DDoS attacks) and test the attack detection accuracy of the framework. The experimental results show that the attack detection accuracy of FL-HCPS only decreases by 2.1% under model poisoning attacks, 1.8% under data tampering attacks, and 1.5% under DDoS attacks. In contrast, TH-FL and PP-FL have a decrease of more than 8% under model poisoning attacks. This indicates that FL-HCPS‘s attack-aware adaptive defense mechanism can effectively identify and resist various attacks, ensuring the robustness of the framework.

6. Discussion

6.1 Limitations of the Current Research

Although the proposed FL-HCPS framework has achieved good performance in experimental evaluations, there are still some limitations that need to be addressed in practical applications: (1) The current framework assumes that the edge cluster head node is trusted, but in actual cloud-edge environments, the cluster head node may be compromised by malicious attacks, leading to the leakage of aggregated model parameters. (2) The privacy-enhanced algorithm combines differential privacy and homomorphic encryption, which increases the computational overhead of edge nodes to a certain extent, and it is difficult to apply to ultra-resource-constrained edge devices (such as wireless sensors with limited battery power). (3) The attack-aware adaptive defense mechanism currently supports the identification of 11 common attack types, but it lacks effective detection and defense capabilities for emerging unknown attacks (such as zero-day attacks). (4) The framework does not consider the impact of network congestion on model parameter transmission, which may lead to delays in model aggregation and update in high-traffic

cloud-edge environments.

6.2 Future Improvement Directions

To address the above limitations and further enhance the practical value of FL-HCPS, future research will focus on the following refined directions: (1) Propose a trust-aware hierarchical federated learning mechanism that introduces a blockchain-based trust evaluation model to evaluate the trustworthiness of cluster head nodes. For untrusted cluster head nodes, a multi-cluster cross-validation mechanism is adopted to ensure the security of model aggregation. (2) Design a lightweight privacy-enhanced algorithm based on model compression and lightweight encryption. Use model pruning and quantization technology to reduce the amount of model parameters, and adopt lightweight encryption algorithms (such as AES-128) to replace homomorphic encryption, reducing the computational overhead of edge nodes. (3) Integrate a zero-shot learning-based unknown attack detection model that uses global threat intelligence to generate attack feature representations, realizing the detection of unknown attacks without labeled data. (4) Explore a network congestion-aware model parameter transmission strategy that uses predictive models to estimate network traffic and adjust the transmission rate and time of model parameters, ensuring the timeliness of model aggregation. (5) Conduct large-scale field tests in industrial IoT and smart city scenarios to verify the scalability and practical applicability of the framework. Collect real-world attack data to optimize the attack detection model and adaptive defense strategy. (6) Study the combination of federated learning and digital twin technology to construct a virtual mirror of the cloud-edge security system, realizing the simulation and prediction of attack evolution and improving the proactive defense capability of the framework.

7. Conclusion

Aiming at the problems of severe data privacy leakage risks and weak security defense capabilities in cloud-edge computing, this study proposes a Federated Learning-Driven Hierarchical Cloud-Edge Collaborative Privacy-Preserving and Security Defense Framework (FL-HCPS). The framework adopts a two-level hierarchical federated learning architecture to realize collaborative training of security models between cloud and edge nodes, reducing communication overhead and improving training efficiency. A privacy-enhanced federated learning algorithm combining differential privacy and homomorphic encryption is designed to protect the privacy of local data and model parameters. An attack-aware adaptive defense mechanism is integrated to dynamically adjust defense strategies according to attack types and intensity, improving the adaptability of the framework to complex attack environments.

Experimental evaluations based on two real-world datasets (EdgeIIoTset and CSE-CIC-IDS2018) show that FL-HCPS achieves an average attack detection accuracy of 96.8%, reduces the data privacy leakage risk by 78.3% compared with the traditional centralized framework, and shortens the model training time by 41.2%. The communication overhead of FL-HCPS is only 23.5% of the traditional horizontal federated learning framework, and the resource utilization rate of edge nodes is significantly improved. The research results demonstrate that FL-HCPS can effectively balance the requirements of privacy protection, security defense, and computing efficiency in cloud-edge computing, providing a new technical solution for the secure and privacy-preserving operation of cloud-edge integrated systems.

In the future, we will further optimize the trustworthiness and lightweight of the FL-HCPS framework, enhance the detection capability of unknown attacks, and promote its application in large-scale industrial IoT and smart city scenarios. We believe that federated learning-driven cloud-edge privacy protection and security defense will become an important development direction of cloud-edge security, providing strong

support for the digital transformation of various industries.

References

[1] Chen, Y., et al. (2023). Homomorphic encryption-based federated learning for privacy-preserving in cloud-edge computing. *IEEE Transactions on Cloud Computing*, 11(3), 2890–2903.

[2] Cloud Security Alliance (CSA). (2025). Global cloud-edge security report 2025. Wakefield, MA: Cloud Security Alliance.

[3] Kowalska, A., et al. (2024). EdgeIIoTset: A comprehensive dataset for industrial IoT edge security. *IEEE Transactions on Industrial Informatics*, 20(5), 5678–5689.

[4] Li, J., et al. (2023). Secure multi-party computation-based federated learning for cloud-edge data sharing. *Journal of Network and Computer Applications*, 215, 103456.

[5] Li, M., et al. (2025). Federated learning for cloud-edge security: A survey. *ACM Computing Surveys*, 58(12), 1–38.

[6] Liu, X., et al. (2024). Vertical federated learning for cloud-edge medical data analysis. *IEEE Journal of Biomedical and Health Informatics*, 28(3), 1234–1247.

[7] McMahan, B., et al. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 1273–1282.

[8] Patel, R., et al. (2024). CSE-CIC-IDS2018-based malware detection for cloud-edge computing. *Future Generation Computer Systems*, 145, 345–358.

[9] Wang, H., et al. (2023). Differential privacy-enhanced federated learning for edge computing. *IEEE Internet of Things Journal*, 10(8), 6789–6802.

[10] Yang, Q., et al. (2025). Cloud-edge collaborative federated learning for smart city applications. *IEEE Transactions on Services Computing*, 18(4), 1987–2000.

[11] Zhang, Y., et al. (2023). Resource-aware federated learning scheduling for cloud-edge computing. *IEEE Transactions on Parallel and Distributed Systems*, 34(7), 2134–2147.

[12] Zhao, Z., et al. (2024). Centralized cloud-edge security defense framework based on machine learning. *Computers & Security*, 130, 103289.

[13] TensorFlow Federated. (2024). TensorFlow Federated 0.60.0 documentation. Retrieved from https://www.tensorflow.org/federated/docs

[14] EdgeIIoTset Dataset. (2024). Warsaw University of Technology. Retrieved from https://www.et.put.pw.edu.pl/~akowalska/datasets/edgeiiotset.html

[15] CSE-CIC-IDS2018 Dataset. (2024). Canadian Institute for Cybersecurity. Retrieved from https://www.unb.ca/cic/datasets/ids-2018.html