EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

Hierarchical Federated Learning for Privacy-Aware Transmission Line Defect Detection

Chong Wang[1,3], Chaoyang Qu[2], Guang Huo[2,*], Qinxuan Chen[2], Shimin Liu[3], Jing Zhang[3], Yongming Li[3]

[1]School of Electrical Engineering, Northeast Electric Power University, Jilin, Jilin, China, 132012
[2]School of Computer Science, Northeast Electric Power University, Jilin, Jilin, China, 132012
[3]State Grid Eastern Inner Mongolia Electric Power Co., Ltd, Hohhot, Inner Mongolia, China, 010020

Abstract

INTRODUCTION: Transmission line defect detection is one of the most conventional and fundamental maintenance tasks in power grids. Its performance in terms of detection accuracy, efficiency, and privacy protection directly affects the operational safety of power systems. Federated learning is considered a secure architecture for cross regional defect detection due to its advantages in data privacy and collaborative training.
OBJECTIVES: The current mainstream two-layer federated learning architecture is difficult to balance detection accuracy with computational and communication efficiency in the face of limited on-site resources in the power grid, and cannot meet the requirements of collaborative transmission line defect detection across multiple regions. To address these issues, this paper proposes a circuit defect detection method based on HFL (Hierarchical Federated Learning, HFL).
METHODS: First, a three-layer federated learning architecture is introduced, where an additional edge layer is incorporated to reduce the computational burden and communication load on the central server. Then, a client dynamic allocation method based on two-factor clustering is proposed. By calculating similarity between client models, clients with approximately independent and identically distributed data are assigned to corresponding edge servers, thereby enhancing training efficiency and detection accuracy for both client and edge models.
RESULTS: Compared to the two-layer federated learning model, the proposed algorithm reduces the computational cost and communication load by 66.67% and 66.86% respectively, while maintaining accuracy and exhibiting stronger generalization ability.
CONCLUSION: The proposed HFL with dynamic client grouping effectively addresses the challenges of data privacy and resource constraints in power grid scenarios. It successfully balances detection accuracy with computational and communication efficiency, offering a practical and scalable solution for collaborative transmission line defect detection across multiple regions.

Keywords: Federated Learning, Privacy Protection, Hierarchical Architecture, Transmission Line Defect Detection

Received on 03 February 2026, accepted on 14 May 2026, published on 28 May 2026

doi: 10.4108/ew.11817

1. Introduction

As the proportion of clean energy continues to rise, the large-scale grid integration of fluctuating power sources such as wind and solar power has become increasingly commonplace. This trend has intensified load fluctuations on transmission lines, simultaneously elevating defects related to aging infrastructure and overheating. Rapid and accurate detection of safety hazards in transmission line infrastructure—including insulators, vibration dampers, and spacer rods—has thus become a critical measure for

*Corresponding author. Email: yanhuo1860@126.com

preventing transmission line failures and ensuring power system security [1]. However, due to the extensive distribution of overhead transmission lines and complex regional environments—such as mountainous areas, lakes, and ditches—manual inspections suffer from low efficiency, high costs, and poor safety, making them inadequate for routine monitoring needs [2]. Consequently, inspection methods have shifted toward drone-based surveys, which not only provide an overview of the overall situation but also capture local details [3]. This involves utilizing artificial intelligence-related technologies to promptly identify safety hazards from the vast amount of inspection images captured.

In recent years, thanks to the rapid advancement of computer vision, deep learning-based detection methods have become the mainstream approach for defect detection and are widely applied in the power industry [4-5]. Compared to traditional image processing methods, deep learning-based methods offer significant advantages in robustness and generalization. Current mainstream algorithms primarily include two-stage detection algorithms based on R-CNN [6-7] and single-stage detection algorithms based on YOLO [8]. The YOLO series of single-stage object detection algorithms proposed by Redmon et al. [9] achieve high detection efficiency and real-time performance by end-to-end regression of object location and category information, demonstrating good adaptability in complex scenarios. Subsequently, Bellou et al. [10] applied YOLO networks to transmission line images from drone inspections, achieving high detection accuracy for typical objects like insulators and fittings. However, such methods inherently rely on large amounts of annotated data for centralized training—a requirement unattainable in power systems. When training data is constrained in scale, category diversity, or distribution, the model's generalization capability significantly deteriorates. In practical applications, power inspection images may contain confidential or sensitive information, making academic or commercial sharing between individuals or third-party institutions challenging. Consequently, there is a lack of publicly available datasets comparable to those in other AI domains. Furthermore, data privacy regulations restrict the exchange of inspection data between power companies. Due to geographical differences and variations in data collection capabilities, inspection data collected from a single region is limited in both quantity and variety, failing to meet the requirements for training highly accurate detection models with strong generalization capabilities. Inspection data collected by power companies in different regions is stored as “data silos” on their respective servers. For data security reasons, it is not feasible to aggregate inspection data from all regions for centralized training of DL models.

To enable collaborative training across regions, “FL” (Federated Learning, FL) technology—designed for distributed data privacy protection and collaborative analysis—has emerged and gradually become a focal point of research [11-12]. This technology achieves collaborative optimization by exchanging model parameters rather than raw data, thereby ensuring that data remains within its domain. This characteristic aligns closely with the privacy protection requirements for power inspection data.Building upon federated learning, Li et al. [13] proposed FedProx, an improved algorithm for FedAvg, to mitigate the impact of heterogeneous client data and computational capabilities on training stability. He et al. [14] extended federated learning to object detection tasks, validating the feasibility of training detection models under multi-source distributed data conditions. Hegiste et al. [15] introduced the Federated Ensemble YOLOv5 method, integrating federated learning with the YOLO detection framework to collaboratively train object detection models across multiple clients, thereby enhancing cross-domain generalization capabilities to some extent. Although these approaches partially alleviate data silo issues, most adopt a two-tier client-server architecture. This leads to significantly increased communication overhead and computational costs on the central server when the number of clients grows or model scale expands. To address practical challenges such as imbalanced distribution of transmission line image data and complex environmental interference, there is an urgent need to design a federated learning architecture featuring dynamic weight allocation and privacy-aware compression communication protocols. This will support the construction of high-precision, low-latency distributed image detection systems, ultimately enhancing the intelligence level and operational efficiency of power inspection. Thus, federated learning-based transmission line defect detection represents an inevitable and optimal technical choice, deeply rooted in understanding and responding to the operational characteristics, data properties, and security requirements of the State Grid. It not only resolves the technical tension between data utilization and privacy protection but also holds significant practical and strategic value for enhancing grid security, improving operational efficiency, and driving digital transformation.

Currently, defect detection for transmission lines under actual operating conditions still faces several key challenges:

1) The accuracy of detection models is highly dependent on the quality of the sample data: power transmission lines span vast areas with significant variations in topography and climate, and defects in different regions exhibit distinct regional characteristics. This results in a highly uneven distribution of defect samples, and data heterogeneity is the primary bottleneck hindering the implementation of federated learning in power detection scenarios.

2) Data privacy protection concerns: Power inspection images collected in practice may contain confidential or sensitive information, making it impossible to establish a unified cross-regional defect dataset through data sharing. Consequently, training a universal transmission line defect detection model is challenging.

3)Limitations in device technical performance: Currently, inspection equipment used by power inspection departments generally lacks high computational capabilities. As the number of client devices increases, the

aggregate computational load and communication traffic on the central server grow linearly. Power inspection terminal devices struggle to handle the communication overhead associated with large-scale federated training, and existing work has yet to propose effective architectural-level solutions tailored to resource-constrained power inspection scenarios.

To address these challenges, this paper proposes a transmission line defect detection method based on hierarchical federated learning. By constructing a unified joint detection model that approximates IID (Independent and Identically Distributed, IID) data, it enables defect detection across regions with heterogeneous data.

The contributions of this paper can be summarized as follows:

(1) We propose a hierarchical federated learning architecture tailored for resource-constrained environments, enabling cross-region collaborative training while ensuring local data remains within its domain. The hierarchical architecture reduces computational scale and communication load on the central server by introducing edge servers positioned between clients and the central server. This facilitates privacy-preserving cross-region collaborative training under limited computational resources.

(2) It introduces a client dynamic admission method based on dual-factor clustering. This approach consolidates clients with initially different distributions into groups that are approximately independent and identically distributed, assigning them to corresponding edge servers. This significantly reduces local heterogeneity at each edge server, thereby accelerating model convergence and enhancing the overall accuracy of the hierarchical federated learning model.

This three-tier architecture forms an elastic collaborative network that preserves the efficiency of asynchronous communication while reducing computational pressure and communication load on the central server through edge-layer buffering. This approach demonstrates greater adaptability when handling federated learning tasks involving heterogeneous devices and unstable network environments.

2. Hierarchical Federated Learning-Based Method for Transmission Line Defect Detection

2.1. Overall Framework of the Model

The hierarchical federated learning architecture consists of the following components, as illustrated in Fig.1.

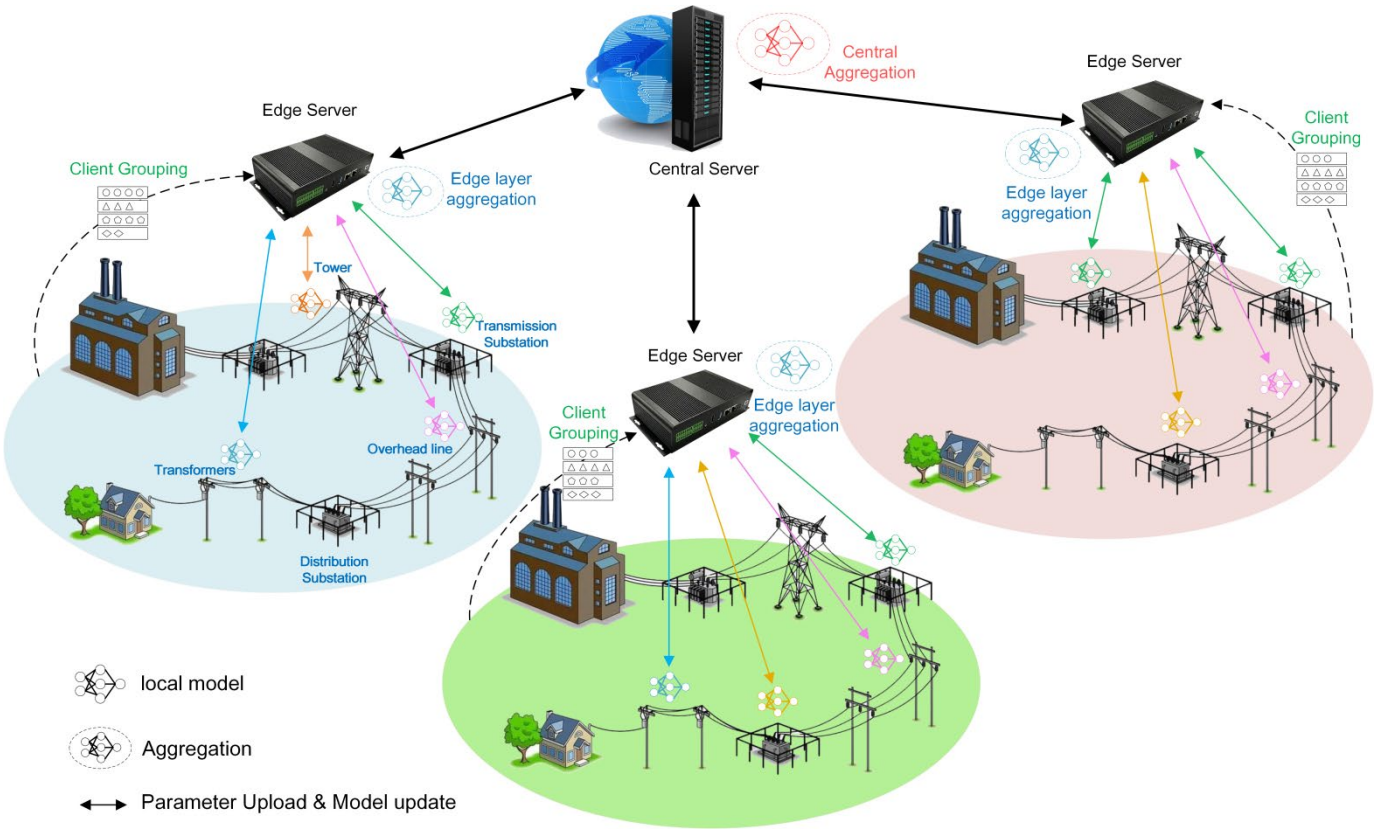

Figure 1. Hierarchical Federated Learning Architecture

(1) Client Model

Transmission line defect detection must be performed at every stage of power transmission, with distinct defect types and characteristics at each stage. This paper defines a detection model applied to a specific transmission segment as a client model. Each client model contains its own local raw data and shares the same defect detection model.

(2) Defect Detection Method

The proposed hierarchical federated learning architecture is a universal framework applicable to mainstream deep learning-based defect detection methods, not limited to the YOLOv8n detection method used in this paper.

(3) Edge Servers

To reduce communication load and computational pressure on the central server, an edge server is established for each region. These edge servers train client models within their respective jurisdictions and aggregate training parameters before submitting them to the central server. Optimizing client grouping on edge servers can improve the overall performance of the training model.

(4) Central Server

The central server receives parameters uploaded from edge servers across regions for global aggregation. It then distributes the updated model to edge servers, which subsequently push it to client models.

Throughout the hierarchical federated learning process, raw image data stored locally by each department remains unshared externally. Only trained model parameters are transmitted, ensuring a degree of data privacy.

2.2. Hierarchical Federated Learning Architecture

In practical power application scenarios, an increasing number of power supply companies are adopting intelligent methods such as robots, unmanned aerial vehicles, and video surveillance equipment for automated power inspections. This generates vast amounts of inspection imagery that requires further processing, recognition, and classification to promptly identify potential hazards and issues from the image data [16]. The two-layer federated learning architecture faces significant communication costs and computational resource pressures. A core challenge for two-layer federated learning is how to reduce communication load while maintaining model accuracy, enabling large-scale deployment even with centrally located servers that have limited computational capabilities.

This paper proposes a three-layer federated learning architecture that optimizes data transmission paths by introducing edge servers between local clients and the central server. Specifically, similar to two-layer federated learning, local clients download global model parameters, execute multiple rounds of SGD (Stochastic Gradient Descent, SGD) using local data to update local models, and compute local weight update vectors (1):

$$\Delta\theta_i^{t+1} = \mathrm{SGD}_k(\theta^t, D_i) - \theta^t \quad (1)$$

Among them, D_i denotes the local data of client i, K represents the number of local SGD iterations performed by the client during a single round of communication, θ^t represents the global model parameters at round t, and $\Delta\theta_i^{t+1}$ denotes the model update of client i at round t+1.

After updating locally, the client sends the updated model to its assigned edge server. Preliminary aggregation is performed on the edge server to obtain the edge model parameters θ_{edge} (2):

$$\theta_{\text{edge}} = \sum_{i=1}^{n} \frac{|D_i|}{|D|} \Delta\theta_i^{t+1} \quad (2)$$

Among them, $|D_i|$ represents the data volume of client i, and $|D|$ denotes the total data volume across all clients. The edge server performs weighted aggregation on each client's model update to generate an edge model.

Finally, the edge server uploads the aggregated edge model to the central server. The central server performs final aggregation across all edge servers to obtain the global model (3):

$$\theta^{t+1} = \sum_{j=1}^{N} \frac{|D_j|}{|D|} \theta_{edge,j} \quad (3)$$

Among them, $\theta_{edge,j}$ represents the aggregation model of edge server j.

In summary, the three-layer architecture not only ensures that clients need not explicitly share raw data [17], but also reduces the burden on the central server by introducing edge servers to perform pre-aggregation of model updates at the edge.

2.3. Dynamic Client Grouping Based on Two-Factor Clustering

Mainstream client grouping methods primarily include grouping based on data similarity and computational capability. However, when handling highly heterogeneous data, it is challenging to effectively balance the distribution heterogeneity and computational resource heterogeneity among clients. The weight differences in heterogeneous distributions can hinder the convergence of aggregation schemes [18]. HFL (Hierarchical Federated Learning, HFL) effectively addresses communication costs and performance issues in complex distributed environments by introducing a multi-level tree structure [19]. However, a limitation of edge-based FL is the finite number of clients each server can access, inevitably leading to training performance degradation [20]. Random grouping algorithms neglect local data quality, leaving the three-tier architecture facing the challenge of how to partition client groups[21].To address the aforementioned issues, this section proposes a client grouping algorithm based on similarity and contribution: by calculating the cosine similarity between client model updates to measure the consistency of data distribution, while quantifying the marginal contribution of each client to the global model,

clients with high similarity and high contribution are assigned to different edge servers. This approach enhances the convergence efficiency of model aggregation while ensuring data diversity within each group.

Specifically, for n clients and M edge servers, after one round of local training by clients, model updates are uploaded to the central server for similarity computation (4):

$$\mathrm{sim}(i,j)=\frac{\Delta\theta_i^t\cdot\Delta\theta_j^t}{\|\Delta\theta_i^t\|\ \|\Delta\theta_j^t} \tag{4}$$

Among them, n ∈ [20, 100], M ∈ [4, 10], and $\Delta\theta_i^t$ and $\Delta\theta_j^t$ represent the model updates for clients i and j, respectively. The Shapley value approximation method is employed to evaluate the marginal contribution of each client's historical updates to the improvement in global model performance. The Shapley value ϕ_i for client i is defined as(5):

$$\phi_i=\sum_{S\subseteq N,\ \{i\}}\frac{|S|!(N-|S|-1)!}{N!}[V(S\cup\{i\})-V(S)] \tag{5}$$

Among them, V represents the global model performance, N denotes the set of clients, and S is a subset of N\{i}, i.e., a subset consisting of all clients except client i. The precise calculation of Shapley values requires summing over all possible subset combinations, computing the marginal contribution [V(S∪{i})−V(S)] for each subset. To group clients, the central server must also compute scores weighted by similarity and contribution (6):

$$G_i=\alpha\cdot\left(\frac{1}{1+\sum_{j\in S}\sin(\Delta\theta_i^t,\Delta\theta_j^t)}\right)+\beta\cdot\phi_i \tag{6}$$

Among them, $\sum_{j\in S}\sin(\Delta\theta_i^t,\Delta\theta_j^t)$ represents the total similarity score between client i and all other currently assigned clients. The summation 1+∑ ensures that clients with higher similarity receive higher scores. ϕ_i represents client i's Shapley value contribution, reflecting its contribution to the global model. α and β are the weighting coefficients for similarity and contribution, respectively. All clients are then ranked in descending order based on their weighted scores G_i. Clients are then sequentially assigned to different edge servers. For example, the client with the highest score is assigned to the first edge server, the second-highest to the second edge server, and so on. This ensures that clients with high similarity and high contribution are distributed across different edge servers. This approach maximizes the diversity of defect types across each edge server, guaranteeing that edge servers are approximately independent and identically distributed.

2.4. Two-stage synchronous aggregation strategy

This study employs the most standard FedAvg algorithm as the model aggregation strategy. This approach synchronizes model updates between the central server and edge nodes, ensuring consistency and comparability of the aggregation process across both two-layer and three-layer architectures. In the two-layer architecture, each client k trains its model using local data, obtains a set of model weights, and uploads them to the central server. The central server performs a weighted average of all client models to form a global model, which is then distributed to each client. The aggregation formula for the central server is shown in formula (7):

$$\theta_{\text{global}}=\frac{1}{N}\sum_{k=1}^{N}\theta_k \tag{7}$$

Among them, N represents the number of clients participating in this training round, and θ_k denotes a set of model weights. Under a three-layer architecture, a hierarchical FedAvg algorithm is employed. Each edge server synchronously receives model parameters uploaded by the clients it manages, performs weighted averaging (FedAvg), and uploads the result to the central server. The central server synchronously aggregates all edge servers to form a global model, which is then distributed to all edge servers. Edge servers subsequently distribute this model to individual clients. This ensures the overall model implementation is simple and computationally efficient.

3. Experiments

3.1 Experimental Setup

To verify the feasibility and superiority of the proposed hierarchical federated learning architecture for cross-regional power line defect detection under the current hardware conditions of the State Grid, this experiment deployed the hierarchical federated learning framework on a laboratory server and implemented it using the PyTorch 2.3.0 deep learning framework. The development environment is Anaconda, and YOLOv8n is used as the base object detection model. A three-tier simulation environment comprising 15 clients, 4 edge servers, and 1 central server was constructed on this server. The aforementioned architectural configuration aligns with the organizational structure and equipment configuration of the patrol inspection department of a provincial subsidiary of the State Grid Corporation of China; therefore, the simulation results presented in this paper can, to a certain extent, reflect the performance of the proposed method under actual operating conditions.

Regarding training parameter configuration, the stochastic gradient descent optimizer was employed. The number of training iterations was fixed at 10, with the learning rate set to the default value of 0.01 for YOLOv8n. The batch size was set to 16, and the communication frequency between edge servers and the central server was configured to occur once per global iteration.

3.2 Experimental Dataset

The experimental data consists of images captured by drones during routine inspections of transmission lines. The dataset contains 5,189 images at 640×640 resolution, representing seven defect types: normal transmission lines, faulty transmission lines, insulators, faulty insulators, vibration dampers, faulty vibration dampers, and high-voltage towers. To simulate the uneven distribution of defect samples across different regions, these images were divided into 15 regions based on topographical features. Each region corresponds to a client, with a training-to-validation ratio of 4:1 for each client. As shown in Fig. 2, the proportion of defect types varies significantly across different clients. The data exhibits Non-IID (Non-Independent and Identically Distributed, Non-IID) characteristics and is imbalanced in volume, thereby simulating the uneven data distribution observed among different power companies in real-world scenarios.

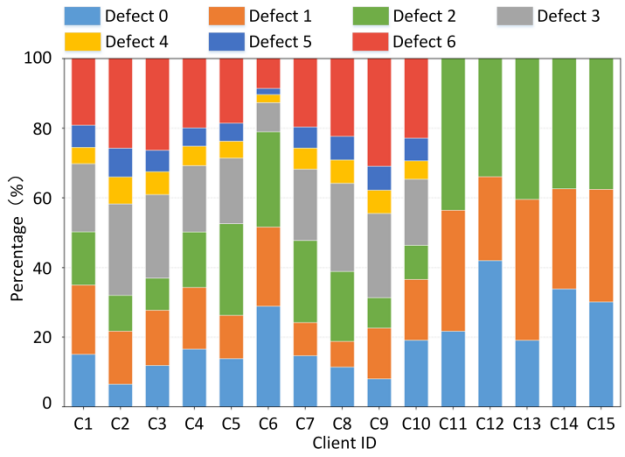

Figure 2. Client Defect Type Distribution

3.3 Evaluation indicators

To demonstrate the effectiveness of the experimental methodology presented in this paper, the following performance evaluation metrics will be employed:

Computational Cost measures the computational resources consumed by an algorithm during execution. It serves as a critical metric for evaluating the efficiency of distributed systems, directly reflecting the algorithm's practical feasibility and resource utilization. Computational cost is typically quantified by the algorithm's time complexity or actual runtime. In experiments, we record the total number of times the central server performs aggregation operations on model parameters (unit: M operations). A lower computational cost indicates higher algorithmic efficiency.

Communication Overhead can be defined as the network expense incurred by data transmission between nodes during collaborative operations, reflecting the system's scalability and network resource consumption. The value of communication overhead equals the total parameters transmitted between nodes (unit: M parameters). A smaller communication overhead indicates higher network efficiency.

Precision represents the proportion of samples correctly predicted as positive among all samples predicted as positive. It is a core metric for measuring accuracy in classification or detection tasks, reflecting the system's ability to avoid false positives. Higher precision indicates lower false detection rates.

Recall represents the proportion of true positives correctly predicted. This is a crucial metric for evaluating a system's detection completeness, reflecting the algorithm's ability to capture target objects. Higher recall indicates lower false negative rates.

The F1-Score is the harmonic mean of precision and recall. This balanced metric comprehensively evaluates system performance by considering both accuracy and completeness of detection results. In experiments, we primarily use the F1-Score as the basis for model selection and parameter tuning. The closer the F1-Score approaches 1, the better the overall system performance.

3.4 Experimental Results and Analysis

3.4.1 Hierarchical Federated Learning Comparison

To fairly compare the performance differences between the two-layer and three-layer architectures, both architectures employ the same YOLOv8n object detection model, with identical global training rounds, local training rounds, and batch sizes. The two-layer architecture utilizes the mainstream FedAvg framework, while the three-layer federated learning, lacking a unified foundational framework, employs standard parameter averaging for aggregation. Clients are partitioned using random grouping.

(1) Computational Cost

The hierarchical aggregation strategy of the three-tier architecture enables distributed processing of computational workloads by offloading partial aggregation tasks to edge servers, significantly reducing computational costs on the central server. As shown in Fig. 3, the horizontal axis represents the number of training iterations, and the vertical axis represents computational cost. The computational costs for the central server under the three-layer architecture and the two-layer architecture are 12.08M and 36.25M, respectively. This represents a significant reduction of 66.67% in computational cost.

(2) Communication Load

Under the three-layer federated learning architecture, the central server communicates only with a small number of edge servers, significantly reducing its communication load. As shown in Fig. 4, the horizontal axis represents the number of training iterations, and the vertical axis represents the communication load. The communication load on the central server is 24.23M and 73.12M for the three-layer and two-layer architectures, respectively. This represents a 66.86% reduction in communication load. Furthermore, as the number of clients increases, the communication load in both two-layer architectures grows

linearly (O(n)), whereas the communication load in the three-layer architecture scales only with the number of edge servers (O(k)), where k << n. This confers a significant scalability advantage to the three-layer architecture in large-scale federated learning scenarios.

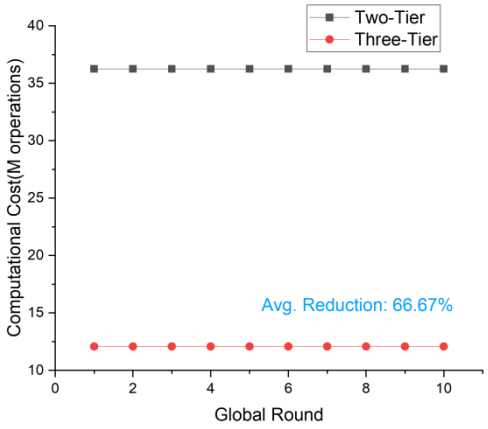

Figure 3. Cost Comparison by Round for Central Server Units

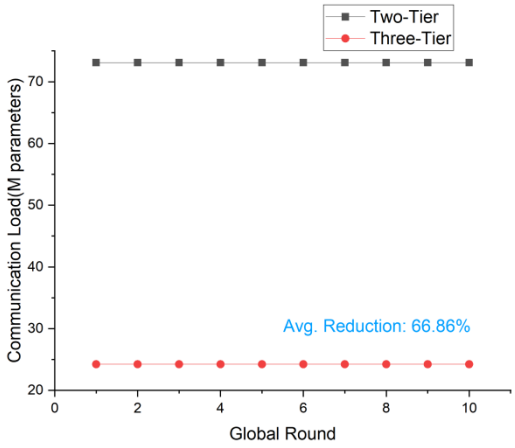

Figure 4. Comparison of Communication Load by Round in Central Server Units

(3) Detection Performance

Fig. 5 and Fig. 6 present line charts showing the trends in precision and recall over the course of 10 rounds of global training. The horizontal axis represents the training round (rounds 1–10), while the vertical axes show the values for precision and recall (range 0–1). The detection performance of the two architectures is evaluated by observing the fluctuations in the two curves. Fig. 5 shows that the three-layer architecture exhibits a slight 0.8% decrease in precision compared to the two-layer architecture, indicating a minor difference. Fig. 6 shows a 4.9% decrease in loss. This demonstrates that the aggregation operation at the edge layer effectively preserves the key features of the client-side model without significantly affecting the quality of the global model.

Furthermore, the three-layer architecture exhibits smaller performance fluctuations across rounds, demonstrating superior stability. This is primarily due to the edge servers acting as intermediate aggregators, mitigating the impact of client model heterogeneity on the global model.

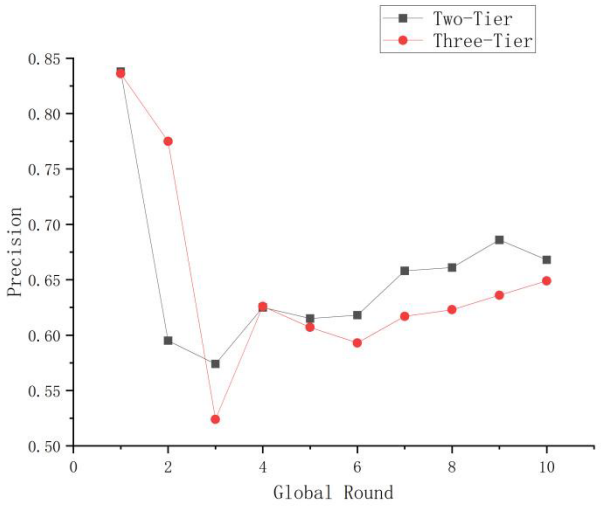

Figure 5. Precision Comparison

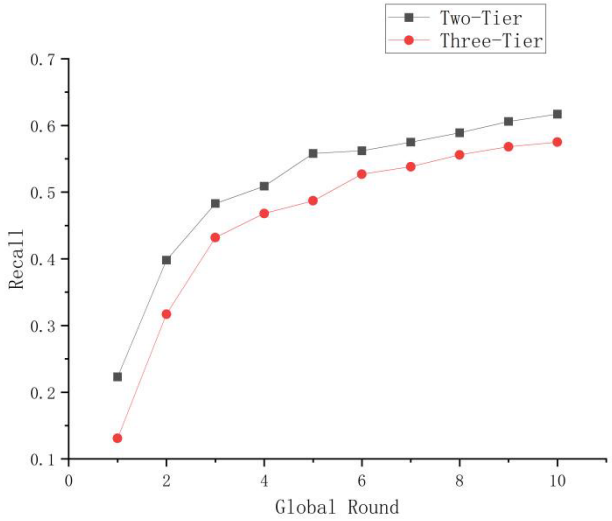

Figure 6. Recall Rate Comparison

Table 1 provides a comprehensive comparison of the two architectures. The results indicate that introducing a three-tier federated learning architecture in real transmission line image detection can be applied on central servers with limited computational resources while ensuring accuracy, effectively meeting the real-world scenarios of collaborative training among multiple power companies.

Table 1. Comparison of Two Architectures

Comparison criteria	Two-layer architecture (e.g., FedAvg)	HFL	Conclusion
Communication Mode	Clients communicate directly with the central server	Introducing edge servers for intermediate aggregation	Significantly reduces central communication load
Communication complexity	O(n), scales linearly with the number of clients	O(k), where k is the number of edge nodes (k≪n)	Significantly improves scalability
Communication overhead	73.12 M	24.23 M	↓66.86%
Training stability	Susceptible to heterogeneous client data	Edge-layer buffer heterogeneity	Less volatility in the layered architecture
Heterogeneous Data Adaptability	Weak (direct aggregation)	Stronger (local aggregation buffer differences)	Better suited for real-world power scenarios
Model accuracy	Benchmark accuracy	Slightly decreased	Accuracy remains largely unchanged

3.4.2 Hierarchical Federated Learning Comparison

This experiment validated the effectiveness of dynamic client grouping by comparing random grouping strategies within a three-layer framework. Table 2 compares the impact of different numbers of edge servers and different weight parameters α on model performance; we conducted a sensitivity analysis by varying $\alpha \in \{0.2, 0.6, 0.9\}$. As shown in Table 2, $\alpha = 0.6$ yields the best performance in terms of both recall and precision. When α is small (e.g., 0.2), the grouping process is primarily driven by model similarity, which may result in overly homogeneous client clusters, thereby limiting the diversity of contributions. Conversely, when α is large (e.g., 0.9), the clustering process overemphasizes customer contributions, which may introduce bias due to heterogeneity in data distribution among customers. Therefore, $\alpha = 0.6$ achieves a balanced compromise between similarity and contribution, yielding more stable and effective clustering results.

Table 2. Comparison under different values of edge servers and weight parameter α

Parameters	value	Recall	Precision	Analysis of Results
Number of edge servers M	2	0.51	0.53	Load concentration, performance limitations
	4	0.54	0.56	Optimal
	8	0.52	0.54	A slight decline due to scattered data
Weighting factor α	0.2	0.50	0.52	Similarity bias
	0.6	0.54	0.56	Optimal
	0.9	0.52	0.54	Contribution bias

Fig. 7 and Fig. 8 are stacked distribution charts. The horizontal axis represents each edge server (ES0–ES3), and the vertical axis represents the proportion of each defect type. Different colors denote different defect types, with larger blocks indicating a greater number of samples for that defect type. Fig. 7 shows the distribution of defect types across edge servers after random client grouping. It reveals that certain defect types are extremely scarce on edge server ES2. In contrast, Fig. 8 demonstrates that under the dynamic grouping strategy, each edge server contains a relatively complete sample of defect types, exhibiting near-independent and identically distributed characteristics. This promotes more comprehensive feature representation learning across edge models.

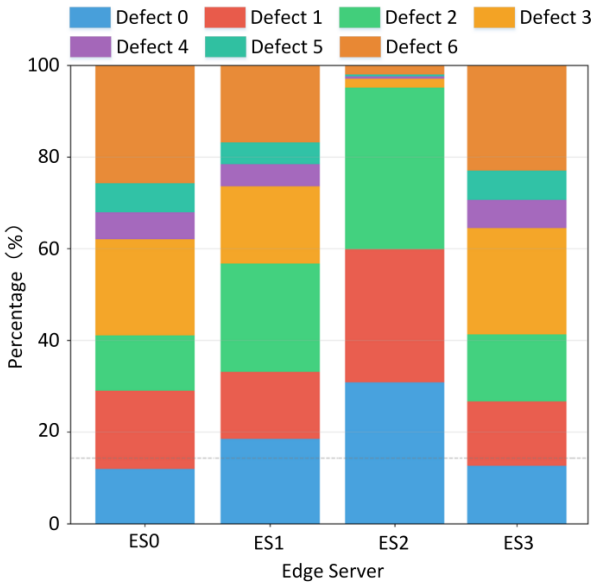

Figure 7. Distribution of Random Grouping Defect Types on Edge Servers

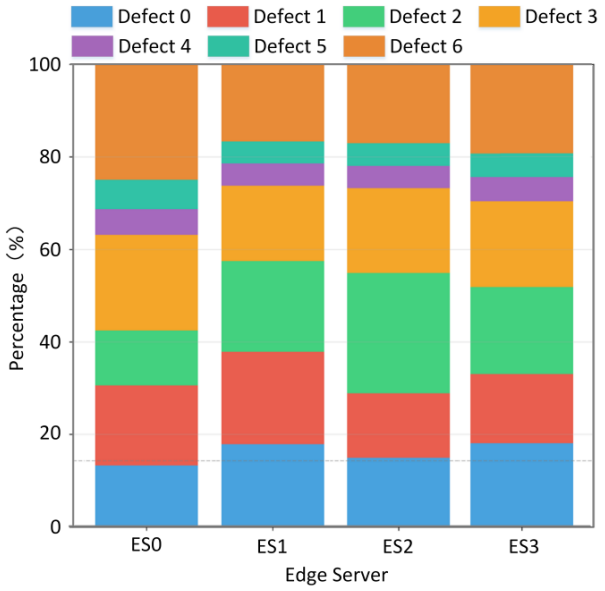

Figure 8. Distribution of Dynamic Grouping Defect Types on Edge Servers

The experiment evaluated client model similarity and data contribution volume, assigning weighted scores to clients and ranking them accordingly, as shown in Fig. 9. the horizontal axis represents each client, and the vertical axis represents the clients' scores; the taller the bar, the higher the score. As indicated by the labels on the bars, the clients with high scores are C9, C7, and C4. Client distribution across edge servers is illustrated in Figs. 10 and 11, where the horizontal axis represents 15 clients and the vertical axis denotes 4 edge servers. Filled boxes indicate clients assigned to that edge server. In Fig. 10, high-scoring clients C9 and C4 are both assigned to the same edge server ES1.

Compared to the random distribution strategy, Fig. 11 demonstrates that the dynamic grouping strategy assigns all high-scoring clients to different edge servers. This indicates that the grouping strategy prevents edge servers from monopolizing high-scoring clients, thereby avoiding data feature clustering.

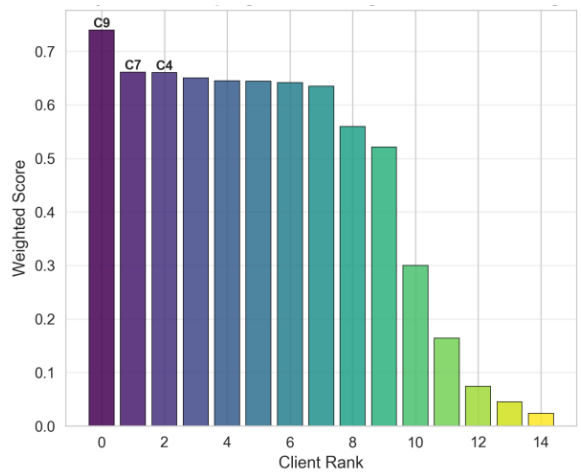

Figure 9. Client Weighted Score Ranking

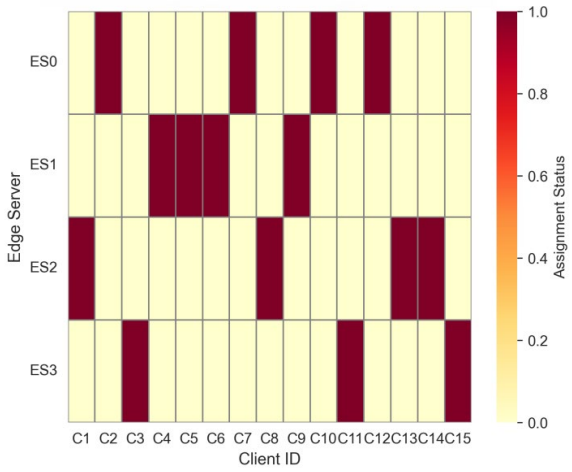

Figure 10. Client Distribution of Random Grouping Strategy

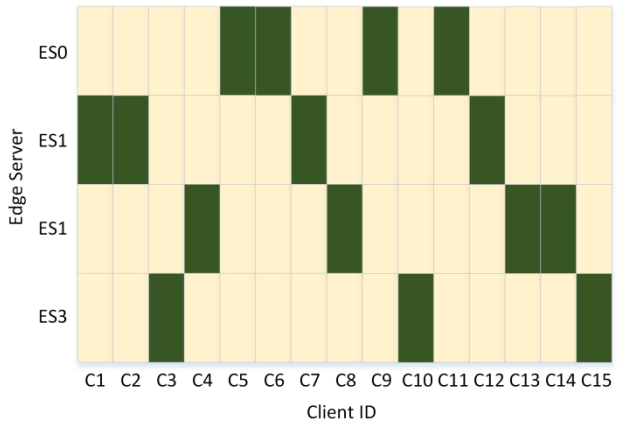

Figure 11. Dynamic Group Policy Client Distribution Status

Beyond the aforementioned experiments, this paper further validates the impact of dynamic grouping on the learning performance of the three-layer federated learning architecture by comparing accuracy convergence curves during training and F1 scores across seven defect categories for both grouping strategies. As shown in Fig. 12, The horizontal axis represents training iterations (1–10), and the vertical axis represents accuracy values. The green line represents dynamic grouping, while the red line represents random grouping. It can be seen that, for the same object detection model, the dynamic grouping strategy significantly outperforms random grouping in terms of both convergence speed and final accuracy.

under the same object detection model, the dynamic grouping strategy significantly outperforms random grouping in both convergence speed and final accuracy. The dynamic grouping strategy exhibits a rapid upward trend during the first five training rounds, primarily

because it better addresses data heterogeneity issues, reducing gradient conflicts during model aggregation. Furthermore, in Fig. 13, the horizontal axis represents seven types of defects, and the vertical axis represents the F1 score. The green bar represents dynamic grouping, and the red bar represents random grouping. The higher the bar, the higher the F1 score. It can be seen that dynamic grouping achieves higher F1 scores across all defect types and has smaller variance in F1 scores for each type of defect, indicating that the model's detection capability for different types of defects is relatively balanced and has better robustness.

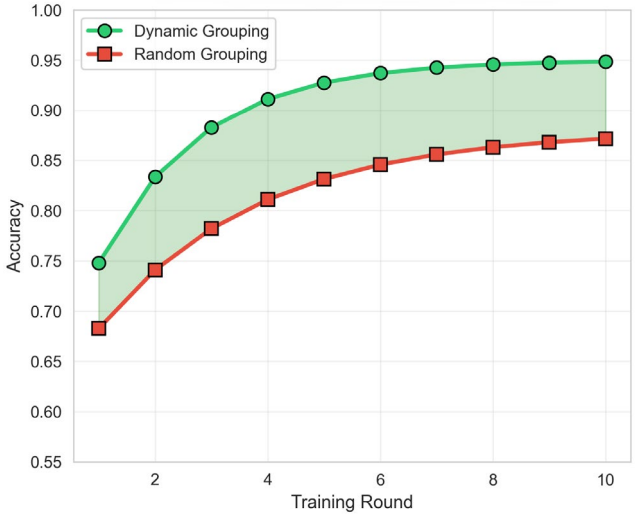

Figure 12. Comparison of Accuracy Between Two Grouping Strategies

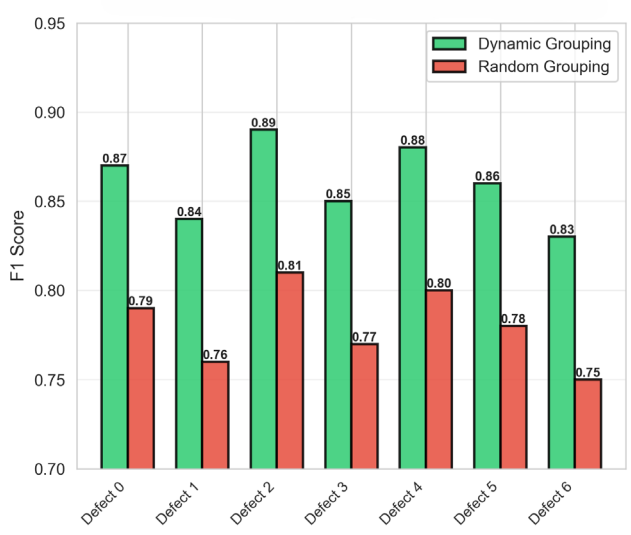

Figure 13. Comparison of F1 Scores for Two Grouping Strategies

4. Conclusion

This paper proposes a hierarchical federated learning transmission line defect detection algorithm based on dynamic client grouping with bi-factor clustering. The method first introduces an edge server layer to alleviate the computational pressure and communication load of the central server, thereby reducing hardware requirements and making it suitable for practical power grid deployment scenarios with limited computing resources. Subsequently, an efficient dynamic client grouping strategy is designed for the three-layer architecture. By adjusting the fault samples within each group to approximate independent and identically distributed, this strategy accelerates model convergence and can effectively support rapid defect localization and response in practical power grid scenarios involving collaborative modeling across multiple substations. The experimental results show that, compared with the two-layer federated learning model, the proposed algorithm reduces computational cost and communication load by 66.67% and 66.86% respectively, while maintaining detection accuracy and demonstrating better generalization ability. This feature makes it particularly suitable for deployment at remote power transmission line monitoring nodes with limited bandwidth.

Although the method described in this paper has achieved satisfactory performance in the task of power line defect detection, several issues remain that warrant further investigation in practical applications. First, complex and variable weather conditions (such as rain, snow, smog, and extreme temperatures) may cause changes in the data distribution across client devices, thereby exacerbating data heterogeneity, affecting the effectiveness of dynamic grouping strategies, and consequently adversely impacting model performance. Therefore, future research could focus on robust grouping methods tailored for strongly non-independent and identically distributed environments, such as introducing adaptive clustering mechanisms or optimizing grouping strategies by incorporating environment-aware information. Second, while the introduction of edge servers effectively reduces communication load, it also increases the potential risk of privacy leakage. Since edge nodes perform intermediate aggregation functions, they may become targets for attacks to some extent. Therefore, it is necessary to incorporate privacy protection mechanisms such as differential privacy, secure multi-party computation, or secure aggregation into the three-layer federated learning framework to enhance the system's security and practicality in complex power grid environments.

Acknowledgements

The authors would like to acknowledge the grant from Science and Technology Project of State Grid East Inner Mongolia Electric Power Company Limited (Grant No. 526603250001).

References

[1] B. Wei, Z. Xie, Y. Liu, et al. Online monitoring method for insulator self-explosion based on edge computing and deep learning. CSEE J. Power Energy Syst. 2021; 8(6):1684–1696.

[2] J. Zhou, X. Liu, Z. Tang, et al. SDH-DETR lightweight insulator defect detection algorithm. Electronic Measurement Technology. 2025; 48(11): 88-104.

[3] T. Tao, Z. Li. A review of deep learning application in transmission line defect detection. Electric Power Systems Research, 2026; 250: 112193.

[4] J. Sun, C. Li, X. Wu, et al. An effective method of weld defect detection and classification based on machine vision. IEEE Transactions on Industrial Informatics. 2019; 15(12):6322-6333.

[5] B. Guo, Y. Wang, S. Zhen, et al. SPEED: Semantic prior and extremely efficient dilated convolution network for real-time metal surface defects detection. IEEE Transactions on Industrial Informatics. 2023; 19(12):11380-11390.

[6] R. Girshick, J. Donahue, T. Darrell, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[7] S. Ren, K. He, R. Girshick, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence. 2016; 39(6): 1137-1149.

[8] B. J. Souza, S. F. Stefenon, G. Singh, et al. Hybrid-YOLO for classification of insulators defects in transmission lines based on UAV. Int. J. Elec. Power.2023; 148:108982.

[9] J. Redmon, S. Divvala, R. Girshick, et al. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[10] E. Bellou, I. Pisica, K. Banitsas. Aerial inspection of high-voltage power lines using YOLOv8 real-time object detector. Energies. 2024; 17(11): 2535.

[11] S. K. Lo, Q. Lu, H. Y. Paik, et al. FLRA: A reference architecture for federated learning systems. European Conference on Software Architecture. Cham: Springer International Publishing, 2021: 83-98.

[12] S. Otoum, N. Guizani , H. Mouftah. On the feasibility of split learning, transfer learning and federated learning for preserving security in ITS systems. IEEE Transactions on Intelligent Transportation Systems. 2023; 24(7): 7462–7470.

[13] T. Li, A. K. Sahu, M. Zaheer, et al. Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems. 2020; 2: 429-450.

[14] C. He, M. Annavaram, S. Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. Advances in neural information processing systems. 2020; 33: 14068-14080.

[15] V. Hegiste, T. Legler, M. Ruskowski. Federated ensemble yolov5–a better generalized object detection algorithm. 2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC). IEEE, 2023: 7-14.

[16] L. Zhong, K. Liu. Visual classification and detection of power inspection images based on federated learning. IEEE Transactions on Industry Applications. 2024; 60(4): 5460-5469.

[17] L. Fan, Z. Zeng, X. Zhang, et al. Joint client selection and epoch configuration for heterogeneity-aware federated learning in resource-constrained systems. Future Generation Computer Systems. 2026;177108249-108249.

[18] H. Qu, S. Zhou. Defect Detection in Transmission Lines Based on Federated Learning of Transfer. Computer Systems Applications. 2024; 33(10): 198-204.

[19] Z. Wu, L. Song, X. Li. A Communication-Efficient Decentralized Hierarchical Federated Learning Architecture. Research on Computer Applications. 2025; 42(05): 1325-1330.

[20] L. Liu, J. Zhang, S. Song, et al. Client-edge-cloud hierarchical federated learning. ICC 2020-2020 IEEE international conference on communications (ICC). IEEE, 2020: 1-6.

[21] W. Lin, Y. Xu, B. Liu, et al.Contribution-based Federated Learning client selection. International Journal of Intelligent Systems. 2022;37(10):7235-7260.