

Article

A Unified Federated Learning Framework for Power Data Terminals Under Privacy and Resource Constraints

Xu Dong ¹, Chang Liu ¹, Jiakai Hao ¹, Yuting Li ¹, Xianzhou Gao ^{2,*}, Ruxia Yang ² and Yujia Zhai ²¹ State Grid Beijing Electric Power Co., Ltd., Beijing 100031, China² China Electric Power Research Institute, Beijing 100192, China

* Correspondence: gaoxianzhou@163.com

Abstract

Power data terminals deployed in smart-grid environments generate massive amounts of operational data, yet the sensitive nature of these data and the existence of cross-region silos make centralized model training difficult in practice. Federated learning offers a natural alternative by enabling collaborative model optimization without transferring raw data, but its direct use in power terminal scenarios is still limited by four coupled challenges: update leakage, malicious or abnormal client behavior, constrained terminal-side resources, and severe Non-IID data heterogeneity. To address these issues, we develop SFL-PDT, a hierarchical federated learning framework tailored to power data terminals. The proposed method is built on a server–edge–terminal architecture. Within this architecture, edge nodes aggregate terminal updates from relatively homogeneous regional groups and perform local robustness screening, while the central server aggregates edge-level updates across heterogeneous regions and coordinates the privacy budget schedule for protected updates. It combines adaptive privacy-aware update perturbation, robust suppression of suspicious regional updates, terminal-oriented update compression, and similarity-guided aggregation for heterogeneous data distributions. Experiments on two representative power-system tasks, load forecasting and fault diagnosis, demonstrate that SFL-PDT achieves a superior overall balance among privacy protection, robustness, efficiency, and predictive performance. Compared with the evaluated baselines, the proposed method more effectively reduces reconstruction-related leakage under different privacy budgets, lowers leakage similarity under gradient inversion attacks, and maintains robust performance when malicious clients participate. It also converges faster and more stably under heterogeneous data partitions. In addition, SFL-PDT achieves the best overall predictive results, reaching an MAE of 0.021 for load forecasting and an accuracy of 88.2% for fault diagnosis, while reducing average terminal-side local training time from 4.3 s to 2.9 s and per-round upload volume from 4.2 MB to 1.5 MB relative to FedAvg. These results suggest that SFL-PDT is a practical solution for secure, efficient, and heterogeneity-aware collaborative learning in power data terminal environments.



Academic Editor: Jen-Hao Teng

Received: 22 April 2026

Revised: 13 May 2026

Accepted: 15 June 2026

Published: 1 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: power data terminals; federated learning; data privacy protection; local differential privacy; robust aggregation; model compression

1. Introduction

1.1. Research Background

As a core infrastructure of the Energy Internet, the smart grid relies on advanced sensing, communication, control, and intelligent analytics to coordinate power gener-

ation, transmission, distribution, and consumption. During this process, power data terminals—such as smart meters, distribution terminals, and electricity information collection devices—continuously produce large volumes of operational data, including load measurements, equipment status records, and fault-related signals. These data constitute an important foundation for downstream intelligent tasks such as load forecasting, equipment health assessment, fault diagnosis, and dispatch optimization.

However, power terminal data are inherently privacy-sensitive. They may reveal user consumption behavior, enterprise operation conditions, and grid-side infrastructure states, and thus are subject to strict security and compliance requirements. Under conventional centralized learning paradigms, raw data from multiple terminals are typically uploaded to a cloud server for joint model training. Such a process not only increases the risk of privacy leakage during transmission and storage, but also faces practical barriers caused by cross-region ownership boundaries and data silos. Federated learning (FL), which enables collaborative model training while keeping raw data on local devices, has therefore become an attractive solution for privacy-preserving intelligence in power systems [1,2].

In recent years, FL has received increasing attention in smart-grid and power-IoT scenarios because it matches the deployment characteristics of widely distributed terminals and privacy-constrained data assets. Nevertheless, directly applying general-purpose FL methods to power data terminals remains nontrivial. In practical terminal environments, strict data sensitivity, heterogeneous local distributions, unstable network conditions, and limited computation and communication resources often appear simultaneously. These coupled constraints make it insufficient to directly transplant standard FL algorithms into realistic power-system deployments.

1.2. Problem Statement and Research Gap

Combined with the deployment assumptions and evaluation settings considered in this paper, three challenges are particularly critical for federated learning over power data terminals.

First, update security and privacy leakage remain major concerns. Although FL avoids direct sharing of raw data, exchanged model updates may still leak sensitive information through model inversion or gradient leakage attacks [3,4]. In addition, some terminals may operate in untrusted or weakly protected environments, making them vulnerable to poisoning behaviors or abnormal update injection. Therefore, power terminal FL requires not only privacy-preserving update transmission, but also robustness against malicious or corrupted participants.

Second, terminal-side resource constraints limit practical deployability. Most power data terminals are embedded or lightweight edge-connected devices with constrained CPU capability, memory capacity, and uplink bandwidth. In such settings, large model updates and repeated communication rounds can easily become the main bottlenecks of deployment. As a result, communication-efficient and computation-aware FL mechanisms are essential for practical power terminal applications [5–7].

Third, strong data heterogeneity degrades convergence stability and generalization. Power data collected from different regions, devices, and operating conditions usually exhibit substantial distribution shifts and sample imbalance. This Non-IID characteristic often causes conventional federated averaging to converge slowly and produce unstable or biased global models [8,9]. For power systems, such instability is particularly problematic because downstream tasks often require both reliable convergence and robust generalization across diverse operating environments.

Existing studies have explored federated learning applications in power systems and have separately investigated privacy protection, robust aggregation, model compression,

and heterogeneous optimization. However, these research directions are often treated as independent modules, whereas power data terminal deployment requires them to be coordinated within the same training pipeline. Relatively few studies jointly consider update-level privacy risks, malicious update suppression, terminal-side efficiency, and Non-IID adaptation in a unified framework. This gap motivates the present work.

1.3. Contributions

To address the above challenges, this paper proposes a secure federated learning algorithm for power data terminals (SFL-PDT). The main contributions are summarized as follows: SFL-PDT is designed as a hierarchical federated learning framework for power data terminals. It integrates privacy protection, robust aggregation, model compression, and heterogeneous-data adaptation within a unified server–edge–terminal architecture.

- We design a three-tier “central server–edge node–terminal device” federated learning architecture for power data terminal scenarios. In this architecture, edge nodes first aggregate terminal updates within relatively homogeneous regional groups, and the central server then aggregates edge-level updates across heterogeneous regions. This organization supports scalable and bandwidth-aware coordination and provides a deployment-friendly structure for large-scale distributed training.
- To mitigate update-level privacy leakage, we introduce a terminal-side protection mechanism based on local differential privacy with adaptive noise injection. The central server coordinates the target privacy budget schedule, and terminals apply the corresponding perturbation before edge aggregation. This design improves resistance to gradient leakage and gradient inversion attacks under the evaluated privacy budget settings.
- To improve robustness against malicious or abnormal participants, we introduce robustness screening during edge-side aggregation and regional suppression at the central server. These steps perform anomaly-aware deviation scoring and weighted attenuation during aggregation, thereby reducing the influence of suspicious updates on the global model.
- To improve deployment efficiency on resource-constrained terminals, we design an adaptive compression strategy that combines structured pruning, sparse transmission, and lightweight quantization, thereby reducing terminal-side computation cost and uplink communication overhead.
- To alleviate the adverse effects of Non-IID regional data, we further develop a similarity-aware weighted aggregation strategy based on distribution summary statistics. At the central server, this module reweights edge-level updates according to their consistency with the global distribution profile, improving convergence stability and model generalization in heterogeneous power terminal environments.

2. Related Work

2.1. Federated Learning in Power Systems

With the increasing demand for privacy-preserving intelligence in smart grids, federated learning has gradually been introduced into a variety of power-system tasks [1,2,10]. Compared with centralized training, FL is particularly attractive in this domain because it enables collaborative modeling without requiring raw operational data to be directly exchanged across regions or organizations.

Existing studies have demonstrated the feasibility of FL in representative power applications. Saputra et al. [11] applied FL to residential energy demand prediction and showed that collaborative learning across multiple grid systems can balance privacy preservation and forecasting accuracy. Zhang et al. [12] developed a federated machinery

and power-equipment fault diagnosis framework, improving model adaptability across different operating conditions through distributed learning. Su et al. [13] studied privacy-aware electric-vehicle charging and smart-community services within an energy-blockchain setting, avoiding direct centralized aggregation of sensitive charging and distribution data. Liu et al. [14] further proposed a dedicated federated learning framework for smart grids to support collaborative analysis of power traces under privacy constraints.

These studies confirm the practical relevance of FL in power systems. However, most existing works remain application-oriented and still largely rely on the canonical FedAvg-style training loop [1]. As a result, several deployment-critical issues for power data terminals are not sufficiently addressed in an integrated manner, including update leakage risks [3,4,15,16], terminal-side resource constraints [5,6,17,18], and strong regional Non-IID data heterogeneity [8,9,19–23]. Taken together, the current literature leaves room for unified federated learning designs that jointly consider privacy leakage, malicious or abnormal participant behavior, deployment efficiency, and heterogeneous regional data in practical power terminal environments.

2.2. Security and Privacy Enhancement Technologies

Security- and privacy-oriented federated learning research mainly focuses on three aspects: secure aggregation and transmission protection, privacy-preserving update perturbation, and robustness against malicious clients.

2.2.1. Secure Aggregation and Transmission Protection

Secure aggregation prevents the aggregator from learning any individual client update by ensuring that only the aggregated result can be recovered. Bonawitz et al. [24] proposed a practical masking-based secure aggregation protocol for cross-device federated learning, while hierarchical schemes such as client–edge–cloud aggregation further reduce uplink traffic and limit direct exposure of raw terminal updates [7]. In broader FL settings, secure aggregation is also commonly discussed together with client-level privacy and robustness requirements [10,25].

For power data terminals with constrained bandwidth and intermittent connectivity, lightweight transmission protection is often more practical than heavy cryptographic deployment alone. In this context, update perturbation, edge-assisted aggregation, and server-side anomaly screening can be combined to improve privacy and robustness while keeping latency and energy overhead within a deployable range [7,24,26,27].

2.2.2. Inference Attacks and Leakage-Resilient Learning

Although federated learning keeps raw data on local devices, exchanged model outputs, gradients, and parameter updates may still reveal sensitive information. Fredrikson et al. [3] showed that model inversion attacks can recover private attributes from exposed model signals. Zhu et al. [4] first demonstrated severe data leakage from shared gradients, and subsequent work such as Geiping et al. [15] and Zhao et al. [16] further showed that high-fidelity or more stable reconstruction is possible under practical settings.

In addition to attack-oriented leakage analysis, formal privacy protection is also widely studied through differential-privacy-based perturbation. Dwork [28] established the theoretical foundation of differential privacy, Abadi et al. [29] demonstrated its integration into deep learning through DP-SGD, and Geyer et al. [25] extended this line to client-level differentially private federated learning. More recently, Noble et al. [30] showed that DP constraints can be integrated with heterogeneity-aware optimization through DP-SCAFFOLD.

Together, these studies suggest that privacy protection in federated learning should be understood not only as raw-data non-sharing, but also as resistance to update-level information leakage during transmission and aggregation. For power data terminals, this issue is particularly important because device updates may travel through wide-area communication links and may be exposed to eavesdropping, interception, or replay in practical deployments. Therefore, leakage-resilient learning in this setting requires mechanisms that reduce update recoverability while remaining compatible with terminal-side resource constraints.

2.2.3. Robust Aggregation Against Malicious Clients

In addition to privacy leakage, federated learning systems must also cope with malicious or abnormal participants. Blanchard et al. [31] studied Byzantine-tolerant distributed learning and inspired robust aggregation rules such as Krum. Yin et al. [32] further analyzed median- and trimmed-mean-based robust distributed learning from a statistical perspective, while Xie et al. [33] proposed Zeno to rank and filter suspicious updates under faulty-worker settings.

Federated settings have also motivated robustness mechanisms tailored to poisoned client updates. Fung et al. [34] investigated Sybil-based poisoning and proposed similarity-aware mitigation. Pillutla et al. [26] proposed robust aggregation for federated learning based on the geometric median. Xie et al. [27] studied certifiably robust federated learning against backdoor attacks, highlighting the need for defense mechanisms with stronger guarantees.

However, most existing approaches focus on one security axis at a time. For power data terminals, privacy leakage, poisoning risk, and deployment constraints often co-exist; therefore, isolated defenses may be insufficient for practical system design.

2.3. Optimization for Resource-Constrained and Heterogeneous Terminals

Power data terminals typically operate under tight computation, memory, and communication budgets. At the same time, their local data distributions are often highly heterogeneous. Existing studies have therefore explored both compression-based efficiency optimization and heterogeneity-aware federated optimization.

2.3.1. Model and Update Compression

Model compression is a widely used strategy for reducing storage, computation, and communication overhead. Han et al. [5] proposed “Deep Compression”, which combines pruning, quantization, and coding to substantially reduce neural network size. For federated communication efficiency, Alistarh et al. [6] proposed QSGD, and Reisizadeh et al. [17] further developed FedPAQ, which combines periodic averaging and quantization to reduce communication cost in federated settings.

Beyond update compression, recent studies have also considered system heterogeneity directly at the model level. Diao et al. [18] proposed HeteroFL to support clients with different computation and communication capabilities using heterogeneous local model sizes. These methods provide important building blocks for resource-constrained FL deployment.

2.3.2. Communication-Efficient Learning Under Non-IID Data

Beyond compression, heterogeneity-aware optimization is crucial for stable federated training. Li et al. [8] proposed FedProx to improve optimization stability in heterogeneous networks by regularizing local objectives. Karimireddy et al. [19] introduced SCAFFOLD to correct client drift via control variates. Wang et al. [20] addressed objective inconsistency

under heterogeneous local updates through normalized averaging. Acar et al. [21] proposed FedDyn to align local and global objectives through dynamic regularization.

Recent work has further expanded this line. Li et al. [22] addressed feature-shift Non-IID settings through local batch normalization, and Li et al. [23] proposed model-contrastive federated learning to reduce representation drift. Reddi et al. [35] extended adaptive optimizers such as Adam and Yogi to federated settings, while Karimireddy et al. [36] showed that centralized optimizer behavior can be mimicked more faithfully in heterogeneous FL. For cross-silo heterogeneous devices, Li et al. [37] proposed a computing-power-aware scheduler to improve efficiency and reduce staleness.

These studies suggest that efficient FL for realistic deployments should jointly consider communication budget, local computation burden, and statistical heterogeneity. This requirement is particularly important in power systems, where regional data often differ in both distribution and scale and where terminal-side resources are unevenly available.

Table 1 summarizes the functional coverage of representative federated learning designs and places SFL-PDT in context.

Table 1. Functional comparison of SFL-PDT and representative federated learning designs.

Method Family	Privacy	Robustness	Compression	Non-IID	Hierarchical PDT Fit
FedAvg	No	No	No	Limited	Limited
FedAvg+DP/DP-SCAFFOLD	Yes	Limited	No	Partial	Partial
Krum/RFA	No	Yes	No	Limited	Partial
FedPAQ/HeteroFL/FedPrune	No	No	Yes	Partial	Partial
FedProx/SCAFFOLD/FedNova	No	No	No	Yes	Partial
SFL-PDT	Yes	Yes	Yes	Yes	Yes

FedAvg is used as the basic sample-weighted FL baseline; FedAvg+DP and DP-SCAFFOLD emphasize privacy protection; Krum and RFA emphasize robust aggregation; FedPAQ, HeteroFL, and FedPrune emphasize efficiency-oriented compression; and FedProx, SCAFFOLD, and FedNova mainly address Non-IID optimization.

2.3.3. Summary of Remaining Gap

Taken together, the existing literature provides important building blocks for applying federated learning to privacy-sensitive and resource-constrained environments. However, for power data terminals, current studies still insufficiently address the joint design of leakage resistance, malicious-update suppression, deployment efficiency, and heterogeneity adaptation within one practical framework. In particular, few studies exploit relatively homogeneous regional groups for edge-side aggregation while handling cross-region Non-IID effects at the central server. This remaining gap motivates the unified design adopted in this work.

3. Design of the Secure Federated Learning Algorithm for Power Data Terminals

To address the privacy risks, data heterogeneity, and resource constraints encountered in power data terminal environments, this paper proposes a secure federated learning framework for power data terminals, termed SFL-PDT. The framework follows a three-tier execution structure consisting of a central server, edge nodes, and terminal devices, which is consistent with the deployment logic of distributed power data terminals. In this design, edge nodes aggregate terminal updates within relatively homogeneous regional groups, whereas the central server aggregates edge-level updates across regions with potentially Non-IID data distributions. On this basis, SFL-PDT coordinates four components that are

central to the considered setting: terminal-side privacy-budget-aware update perturbation, edge- and server-side suppression of abnormal updates, terminal-oriented update compression, and similarity-aware aggregation across heterogeneous regions.

Throughout this section, the information exchanged between terminals and the server is represented by model updates. This formulation avoids ambiguity among gradients, parameter increments, and local model differences, and it remains consistent with the update-based aggregation procedure adopted in the experimental evaluation.

3.1. Design of the SFL-PDT Algorithm

3.1.1. Overall Algorithm Framework

The overall structure of SFL-PDT is shown in Figure 1. The framework adopts a three-tier architecture consisting of a central server, edge nodes, and terminal devices. In this architecture, the central server is responsible for global coordination and model updating, edge nodes act as regional coordinators between the server and terminals, and terminal devices perform local training on private data. Terminals managed by the same edge node are assumed to come from a relatively homogeneous region or device group and are therefore treated as IID-like for edge-side aggregation, while heterogeneity is mainly modeled across edge nodes. On this basis, privacy-preserving perturbation, abnormal update suppression, update compression, and similarity-aware aggregation are incorporated into a unified learning framework.

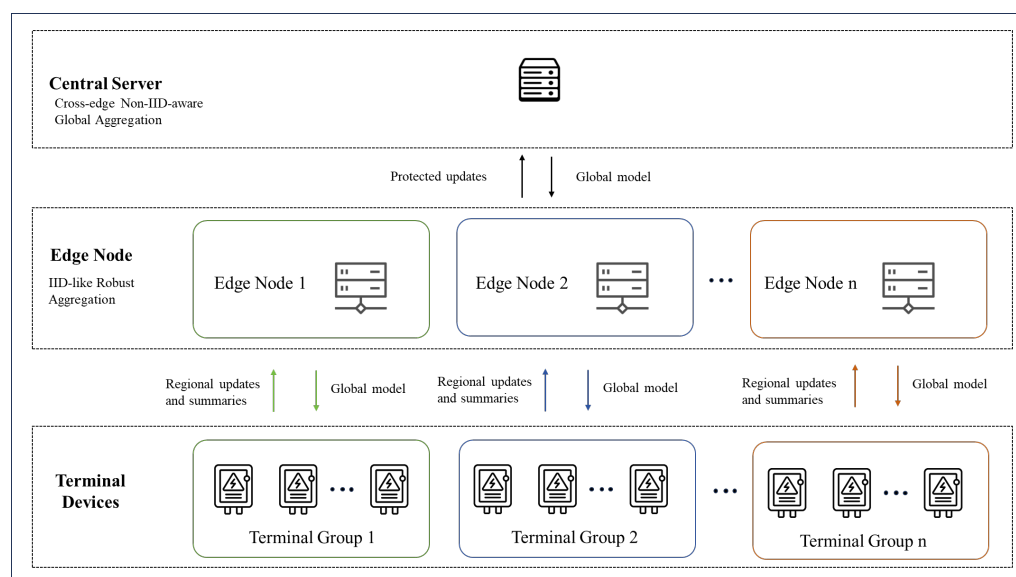


Figure 1. System architecture of SFL-PDT and its two-level aggregation process. Different colors denote different regional edge-node groups. Downward arrows indicate model dissemination, upward arrows indicate update uploading, and ellipsis symbols indicate omitted edge nodes or terminal devices of the same type.

Consider a PIoT system with a set of edge nodes $\mathcal{K} = \{1, 2, \dots, K\}$. Edge node $k \in \mathcal{K}$ manages a set of terminal devices S_k , and terminal i holds a local dataset D_i with sample size N_i . The total number of samples in the system is

$$N = \sum_{k=1}^K \sum_{i \in S_k} N_i. \quad (1)$$

Under the standard federated learning formulation, the global model is trained by minimizing the sample-weighted empirical risk

$$\min_{\theta} F(\theta) = \sum_{k=1}^K \sum_{i \in S_k} \frac{N_i}{N} F_i(\theta), \quad (2)$$

where θ denotes the global model parameters and $F_i(\theta)$ is the empirical risk evaluated on the local dataset of terminal i .

Equation (2) describes the underlying learning objective of the system. In practical power terminal scenarios, however, direct sample-size-weighted averaging is often insufficient because local data distributions may differ substantially across regions and some transmitted updates may be strongly deviating or corrupted. SFL-PDT therefore preserves the sample-weighted objective as the basic optimization target, while organizing the actual aggregation process into an edge-side IID-like stage and a central Non-IID-aware stage. Update protection, robustness-oriented suppression, terminal-side compression, and similarity-aware reweighting are then introduced into this training process.

3.1.2. Algorithm Workflow

The workflow of SFL-PDT includes global model dissemination, local update generation and protection, regional update aggregation, robust global aggregation, and iterative synchronization. The overall training process of SFL-PDT is shown in Figure 2. In each communication round, the server first distributes the current global model to the edge nodes, and the model is then forwarded to the participating terminals. Each terminal performs local training, constructs its local update, and applies clipping, perturbation, and compression before uploading the processed update. The edge nodes then perform robust aggregation over terminal updates within each regional group, and the central server further aggregates the resulting protected edge-level updates across heterogeneous regions. The description of each stage is given as follows.

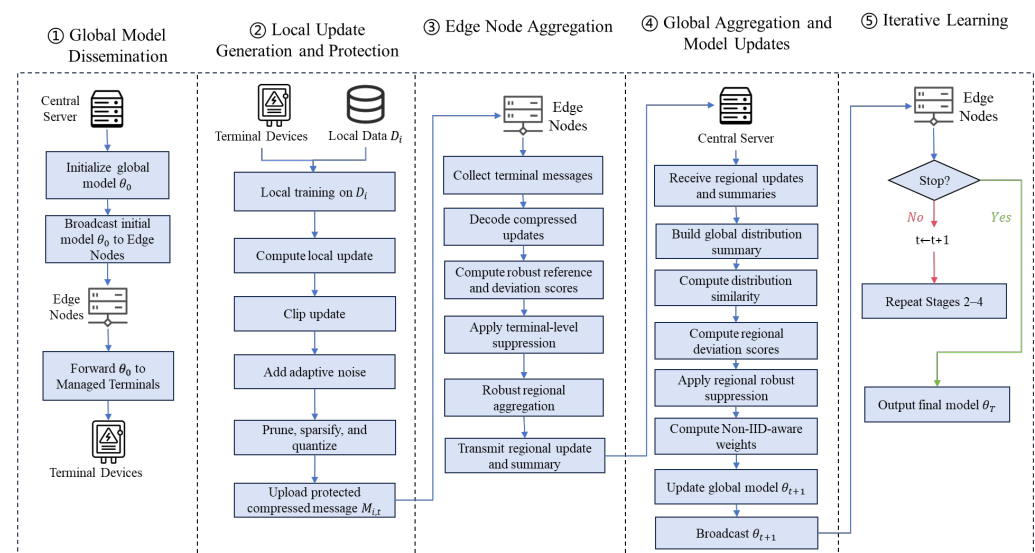


Figure 2. Workflow of the SFL-PDT Algorithm. The dashed rectangle denotes the complete iterative training workflow. Blue arrows indicate the normal execution and communication flow among the central server, edge nodes, and terminal devices, while red and green arrows denote the “No” and “Yes” branches of the stopping criterion, respectively.

1. **Global Model Dissemination Stage.** The central server maintains the current global model θ_t and broadcasts it to the participating edge nodes. Each edge node then

forwards θ_t to the terminal devices selected for the current communication round according to regional connectivity and device availability.

- Local Update Generation and Protection Stage. Each participating terminal i performs E epochs of local training on its private dataset D_i and obtains the local model $\theta_{i,t}^{(E)}$. Based on the current global model and the locally trained model, the terminal-side update is defined as

$$\Delta_{i,t} = \theta_t - \theta_{i,t}^{(E)}. \quad (3)$$

The update is first clipped under an ℓ_2 -norm constraint and then perturbed by the terminal-side mechanism described in Section 3.2.1, yielding the privacy-protected update $\tilde{\Delta}_{i,t}$. To reduce terminal-side computation and uplink communication cost, $\tilde{\Delta}_{i,t}$ is further compressed through the pruning, sparsification, and quantization procedures introduced in Section 3.3. The resulting compressed message is then uploaded to the associated edge node.

- Regional Update Aggregation Stage. After receiving the uploaded messages from terminals in its region, edge node k decodes the compressed updates. Because terminals under the same edge node form a relatively homogeneous regional group, the edge node performs IID-like aggregation while applying local robustness screening to reduce the impact of abnormal terminal uploads. Let

$$N_k = \sum_{i \in S_k} N_i \quad (4)$$

denote the total number of samples managed by edge node k . The edge node first constructs a local robust reference vector and computes a terminal-level suppression coefficient

$$m_{k,t}^{\text{edge}} = \text{Median}\left(\left\{\hat{\Delta}_{i,t}\right\}_{i \in S_k}\right), \quad (5)$$

$$d_{i,k,t}^{\text{edge}} = \left\|\hat{\Delta}_{i,t} - m_{k,t}^{\text{edge}}\right\|_2, \quad (6)$$

$$a_{i,k,t} = \frac{1}{1 + \exp(\gamma_e(d_{i,k,t}^{\text{edge}} - \tau_e))}, \quad (7)$$

where $a_{i,k,t}$ attenuates highly deviating terminal updates within the same edge region. The corresponding edge-level update is then computed as

$$\Delta_{k,t}^{\text{edge}} = \sum_{i \in S_k} \frac{N_i a_{i,k,t}}{\sum_{j \in S_k} N_j a_{j,k,t}} \hat{\Delta}_{i,t}, \quad (8)$$

where $\hat{\Delta}_{i,t}$ denotes the decoded update of terminal i . The aggregated regional update $\Delta_{k,t}^{\text{edge}}$ is subsequently transmitted to the central server. In this way, abnormal terminal updates are attenuated before cross-region aggregation, while regional heterogeneity is handled at the central server.

- Robust Global Aggregation Stage. Upon receiving $\{\Delta_{k,t}^{\text{edge}}\}_{k=1}^K$ from the participating edge nodes, the central server evaluates the consistency of each regional update using the suppression mechanism described in Section 3.2.2 and computes the distribution-similarity weight of each region according to Section 3.4. Combining regional sample size, similarity, and robustness, the final aggregation weight is denoted by $w_{k,t}$. The global model is then updated by

$$\theta_{t+1} = \theta_t - \eta \sum_{k=1}^K w_{k,t} \Delta_{k,t}^{\text{edge}}, \quad (9)$$

where η is the server-side learning rate.

5. Iterative Synchronization Stage. The updated global model θ_{t+1} is broadcast to the edge nodes and then forwarded to the terminals for the next communication round. Stages 2–4 are repeated until a stopping criterion is satisfied, such as reaching the maximum number of communication rounds or achieving stable validation performance. The final global model is then used for downstream power applications, including load forecasting and fault diagnosis.

Overall, the above procedure forms a complete training pipeline of local training, local update protection, compressed transmission, edge-side aggregation, and server-side robust global updating.

3.2. Dual-Layer Security Mechanism Design

SFL-PDT adopts a dual-layer security design. The first layer operates at the terminal side and protects local updates before transmission. The second layer operates during aggregation and suppresses suspicious updates. In this framework, aggregation-side robustness includes both terminal-level screening at the edge nodes and regional suppression at the central server. The terminal-side layer reduces update leakage risk under a target privacy budget setting, whereas the aggregation-side layer improves tolerance to abnormal or corrupted updates in the evaluated training setting.

3.2.1. Privacy-Budget-Aware Protection of Local Updates

To reduce the risk of update leakage and reconstruction attacks, each terminal perturbs its local update before uploading it. Since the magnitude of local updates may vary considerably across terminals and rounds, the update is first clipped under an ℓ_2 constraint:

$$\bar{\Delta}_{i,t} = \Delta_{i,t} \cdot \min\left(1, \frac{C}{\|\Delta_{i,t}\|_2}\right), \quad (10)$$

where $C > 0$ is the clipping threshold. After clipping, the update norm is bounded by

$$\|\bar{\Delta}_{i,t}\|_2 \leq C. \quad (11)$$

This bound provides a unified sensitivity control for subsequent perturbation.

Gaussian noise is then added to the clipped update:

$$\tilde{\Delta}_{i,t} = \bar{\Delta}_{i,t} + z_{i,t}, \quad z_{i,t} \sim \mathcal{N}(0, \sigma_t^2 C^2 I), \quad (12)$$

where σ_t is the noise multiplier at round t and I is the identity matrix. The resulting update $\tilde{\Delta}_{i,t}$ is the privacy-protected version used in the transmission stage.

To balance update protection and learning utility across training rounds, the noise magnitude is gradually reduced over time:

$$\sigma_t = \sigma_{\max} - \frac{t-1}{T-1}(\sigma_{\max} - \sigma_{\min}), \quad (13)$$

where T is the total number of communication rounds, and $\sigma_{\max}$ and $\sigma_{\min}$ denote the initial and final noise multipliers, respectively. In this way, stronger perturbation is applied in the early stage, while later rounds are allowed to use weaker noise to improve convergence behavior.

Given a target privacy budget setting $(\epsilon_{\text{tot}}, \delta_{\text{tot}})$, the sequence $\{\sigma_t\}_{t=1}^T$ can be selected accordingly. The central server distributes the selected noise schedule to the participating terminals, and each terminal perturbs its local update before edge aggregation. In implementation, we use a simple round-wise allocation such as $\delta_t = \delta_{\text{tot}}/T$ and choose the corresponding noise schedule to match the target perturbation level. The privacy budget settings reported in the experiments therefore serve as practical reference points for evaluating the privacy–utility trade-off under the adopted schedule. More refined accounting methods, such as Rényi differential privacy or the moments accountant, can be incorporated in future implementations.

3.2.2. Robust Suppression of Abnormal Regional Updates

Besides update leakage, federated training in power terminal environments may also be affected by abnormal or corrupted updates. After local terminal-level screening at the edge nodes, the central server further examines whether each aggregated regional update is consistent with the dominant update pattern.

Given the set of regional updates $\{\Delta_{k,t}^{\text{edge}}\}_{k=1}^K$, the server first constructs a robust reference vector by coordinate-wise median:

$$m_t = \text{Median}\left(\left\{\Delta_{k,t}^{\text{edge}}\right\}_{k=1}^K\right), \quad (14)$$

where $\text{Median}(\cdot)$ denotes the coordinate-wise median operator. Compared with the arithmetic mean, the median is less sensitive to outliers and is therefore more suitable as a reference for deviation scoring.

The deviation score of edge node k is defined as

$$s_{k,t} = \left\|\Delta_{k,t}^{\text{edge}} - m_t\right\|_2. \quad (15)$$

A larger value of $s_{k,t}$ indicates that the corresponding regional update deviates more strongly from the dominant update trend.

Instead of discarding suspicious updates entirely, SFL-PDT applies a smooth suppression rule:

$$r_{k,t} = \frac{1}{1 + \exp(\gamma(s_{k,t} - \tau))}, \quad (16)$$

where τ is the anomaly threshold and $\gamma > 0$ controls the suppression strength. The sigmoid form provides smooth attenuation: updates close to the dominant trend retain most of their weight, whereas strongly deviating updates are assigned smaller weights without relying on a hard cutoff. This design reduces the contribution of highly deviating updates while avoiding overly aggressive hard filtering, which is particularly useful in heterogeneous settings where a large deviation does not necessarily imply malicious behavior. Under the evaluated setting, this module should therefore be understood as a robustness-oriented suppression mechanism for abnormal regional updates rather than a universal defense against all adversarial behaviors.

3.3. Adaptive Model Compression Strategy

To accommodate the limited computation and communication capabilities of power data terminals, SFL-PDT further compresses terminal-side updates before transmission. The compression module serves as a terminal-oriented efficiency mechanism: it reduces local training cost and uplink payload while preserving the overall update-based learning procedure.

3.3.1. Structured Pruning for Terminal-Side Computation Reduction

Structured pruning is applied to reduce the effective number of trainable structural units during local training. For convolutional models, the structural units can be channels; for recurrent models, they can be groups of hidden units. Let $W_{l,j}$ denote the parameter block associated with the j -th structural unit in layer l . Its importance score is measured by

$$u_{l,j} = \|W_{l,j}\|_2. \quad (17)$$

Units with larger norms are regarded as more important to the current model representation.

Given a pruning ratio $\rho \in [0, 1)$, the binary mask of layer l is defined by

$$m_{l,j} = \begin{cases} 1, & u_{l,j} \in \text{Top-}(1 - \rho) \text{ of layer } l, \\ 0, & \text{otherwise,} \end{cases} \quad (18)$$

that is, only the most important $1 - \rho$ proportion of structural units is retained. The effective trainable parameters in layer l are then written as

$$W_l^{\text{pruned}} = m_l \odot W_l, \quad (19)$$

where $\odot$ denotes element-wise multiplication. This reduces the amount of computation performed during local training while preserving the main structural components of the model.

3.3.2. Sparse Transmission and Quantization

After perturbation, the terminal-side update is further compressed before transmission. Specifically, only the k_t entries with the largest magnitudes are retained. Their index set is

$$I_{i,t} = \text{TopK}(|\tilde{\Delta}_{i,t}|, k_t). \quad (20)$$

Using $I_{i,t}$, the sparse update is defined as

$$\hat{\Delta}_{i,t}[u] = \begin{cases} \tilde{\Delta}_{i,t}[u], & u \in I_{i,t}, \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

This step keeps the most informative coordinates of the update and suppresses the rest.

To further reduce transmission cost, the retained nonzero values are quantized to b bits. Let $x_{\min}$ and $x_{\max}$ denote the minimum and maximum values among the retained entries. The quantization step size is

$$\delta_b = \frac{x_{\max} - x_{\min}}{2^b - 1}, \quad (22)$$

and the corresponding quantization operator is

$$Q_b(x) = \text{round}\left(\frac{x - x_{\min}}{\delta_b}\right)\delta_b + x_{\min}. \quad (23)$$

The final message uploaded by terminal i at round t is therefore

$$M_{i,t} = (I_{i,t}, Q_b(\hat{\Delta}_{i,t}[I_{i,t}])). \quad (24)$$

The edge node decodes $M_{i,t}$ to recover $\hat{\Delta}_{i,t}$ and then performs the edge-level aggregation in Equation (8). In this way, the compression module reduces both terminal-side training time and uplink communication overhead within the same learning pipeline.

3.4. Weighted Aggregation Based on Distribution Similarity

Power data collected from different regions often exhibit significant statistical heterogeneity. To improve the stability of federated optimization under such Non-IID conditions, SFL-PDT incorporates a lightweight similarity-aware aggregation mechanism at the server side. This module uses compact regional distribution summaries to adjust aggregation weights, so that regions more consistent with the global profile can contribute more effectively to model updating.

3.4.1. Construction of Distribution Summary Statistics

Each edge node reports a lightweight summary vector that characterizes the data distribution in its region. For classification tasks, this vector may represent the empirical class-frequency distribution. For regression tasks, it may be constructed from normalized histogram statistics. In general, the summary vector of edge node k is written as

$$q_k = [q_{k,1}, q_{k,2}, \dots, q_{k,M}], \quad q_{k,j} \geq 0, \quad \sum_{j=1}^M q_{k,j} = 1. \quad (25)$$

The corresponding global summary is obtained by sample-size-weighted averaging:

$$q^{\text{global}} = \sum_{k=1}^K \frac{N_k}{N} q_k. \quad (26)$$

This summary provides a compact regional descriptor for similarity-aware aggregation while avoiding direct sharing of raw local data.

3.4.2. Similarity Score Computation

The discrepancy between the regional summary of edge node k and the global summary is measured by the Kullback–Leibler divergence

$$D_k = \text{KL}(q_k \| q^{\text{global}}). \quad (27)$$

A smaller divergence indicates that the corresponding region is more consistent with the overall data profile. Based on this divergence, the similarity score is defined as

$$\phi_k = \exp(-D_k). \quad (28)$$

Hence, edge nodes whose summary statistics are closer to the global profile receive larger similarity weights. KL divergence is used here because the regional summaries are normalized distributions, and it provides a simple way to measure their deviation from the global profile.

3.4.3. Final Similarity-Aware Robust Aggregation

The final aggregation weight combines three factors: regional sample size, distribution similarity, and robustness. Specifically, the weight of edge node k at round t is defined as

$$w_{k,t} = \frac{N_k \phi_k r_{k,t}}{\sum_{j=1}^K N_j \phi_j r_{j,t}}. \quad (29)$$

In this formulation, regions with more samples, higher representativeness, and lower anomaly scores contribute more strongly to the global update.

The server then updates the global model by

$$\theta_{t+1} = \theta_t - \eta \sum_{k=1}^K w_{k,t} \Delta_{k,t}^{\text{edge}}, \quad (30)$$

where η is the server-side learning rate. Equation (30) completes the full aggregation process of SFL-PDT: local terminal updates are first protected and compressed, then aggregated regionally according to sample size and local robustness, and finally reweighted at the server by both regional robustness and distribution similarity.

Overall, SFL-PDT forms an integrated federated learning pipeline for power data terminals under the considered deployment setting. Terminal-side perturbation reduces update leakage risk, robust suppression improves tolerance to abnormal regional updates, compression lowers terminal-side computation and communication burden, and similarity-aware weighting stabilizes training under heterogeneous regional data. Accordingly, the subsequent experiments evaluate the proposed method from four aligned perspectives: update leakage mitigation, robustness to corrupted updates, convergence under heterogeneous client distributions, and terminal-side efficiency.

3.5. Complexity and Scalability Considerations

The computational cost of SFL-PDT mainly comes from terminal-side update processing, edge-side regional aggregation, and server-side aggregation across edge nodes. Let P denote the number of model parameters, M the dimension of the distribution summary vector, K the number of edge nodes, and $|S_k|$ the number of terminals managed by edge node k . Terminal-side clipping and perturbation are linear in the update dimension, and Top- k sparsification can be implemented with partial sorting. The edge-side local robustness screening and aggregation scale with $O(|S_k|P)$ per edge node, while the central robust aggregation over edge updates scales with $O(KP)$. The similarity-aware weighting adds $O(KM)$, which is usually much smaller than update-dimensional operations because $M \ll P$.

The hierarchical structure reduces the aggregation burden on the central server because it receives one protected edge-level update from each edge node instead of all terminal uploads directly. The main communication cost per terminal is reduced from transmitting a dense P -dimensional update to transmitting the retained sparse indices and quantized values. In very large deployments, scalability can be further improved by limiting the number of active terminals per edge node in each round, increasing the number of edge coordinators, or using coarser distribution summaries. These properties make SFL-PDT suitable for hierarchical power terminal deployments and also indicate the directions needed for larger-scale implementations [38].

4. Experimental Analysis

4.1. Experimental Setup

4.1.1. System Setup and Training Protocol

To evaluate the practical applicability of the proposed SFL-PDT framework in power data terminal scenarios, a three-tier federated learning testbed consisting of a central server, edge nodes, and terminal devices was constructed. The central server was deployed on a workstation equipped with an Intel Xeon E5-2680 v4 CPU, 64 GB RAM, and Ubuntu 20.04, and was responsible for global coordination and model aggregation. Five Raspberry Pi 4B boards (4 GB RAM) were used to emulate regional edge nodes, each managing 20 terminal

clients. Terminal devices were emulated under resource-constrained settings consistent with embedded power data acquisition environments. This hierarchical setup follows the proposed “server–edge–terminal” deployment logic and reflects the communication bottlenecks and heterogeneous participation commonly encountered in PIoT scenarios.

For all methods, the same training protocol was used to ensure a fair comparison. In each communication round, all available terminals under the active edge nodes participated in local updating. For the load forecasting task, an LSTM-based predictor was adopted; for the fault diagnosis task, a lightweight CNN classifier was used. The learning rate was set to 0.01, and the maximum number of communication rounds was 100. To reduce reporting variance, each experiment was repeated three times with different random seeds, and the mean results are reported.

4.1.2. Datasets and Non-IID Partitioning

Two public power-related datasets were used for evaluation:

- Dataset 1 (Electric Load Dataset): A regional electric load forecasting dataset derived from the PJM power market [39], containing hourly load measurements and auxiliary environmental variables. The load series was transformed into supervised samples using a 24-step historical window for one-step-ahead prediction.
- Dataset 2 (Equipment Fault Dataset): A public power transformer fault diagnosis dataset [40] containing condition-related features and health labels, which was used to evaluate classification performance in the hierarchical federated setting.

Both datasets were normalized to the $[0, 1]$ interval and split into training, validation, and test sets with a ratio of 7:2:1. The federated setting used 5 edge nodes and 100 terminal clients, with 20 clients under each edge node. Terminals assigned to the same edge node were constructed from the same region or device group and were treated as IID-like within the edge. Cross-edge heterogeneity was then produced by assigning different regions, device groups, or label proportions to different edge nodes. Accordingly, the Non-IID setting was generated through region/device-based partitioning rather than synthetic Dirichlet sampling.

The main partition protocol is summarized in Table 2.

Table 2. Dataset preprocessing and federated partition protocol.

Dataset	Preprocessing	Within-Edge Setting	Cross-Edge Setting
Electric load	Min–max normalization; 24-step historical window; 7:2:1 train/validation/test split	Regionally grouped terminal samples, treated as IID-like within each edge node	Different regional load profiles assigned to different edge nodes
Equipment fault	Min–max normalization of condition-related features; health labels used for classification; 7:2:1 train/validation/test split	Device-grouped terminal samples, treated as IID-like within each edge node	Different device groups or label proportions assigned to different edge nodes

4.1.3. Baseline Methods and Evaluation Metrics

We compared SFL-PDT against four representative federated learning baselines:

- FedAvg: Classical federated averaging without privacy, robustness, or efficiency enhancement;
- FedAvg+DP: Federated averaging with fixed Gaussian perturbation for privacy protection;

- FedPrune: A compressed FL baseline with static pruning-based model reduction, implemented with a fixed magnitude-pruning ratio during local training and communication;
- Krum: A Byzantine-robust aggregation baseline used to reduce the influence of abnormal or malicious updates; in the hierarchical setting, it is applied to terminal updates at each edge node before regional aggregation.

Krum is used as a dedicated robust aggregation baseline in the malicious client experiments, and for fairness it is applied at the same edge-side aggregation stage as the proposed screening mechanism. The evaluation is organized around four aspects: privacy and security, learning effectiveness, deployment efficiency, and module-level contribution. The main metrics include reconstruction accuracy, leakage similarity, fault diagnosis accuracy, load forecasting MAE, terminal-side local training time, and single-round upload volume.

4.2. Privacy and Security Evaluation

4.2.1. Privacy–Utility Trade-Off Under Different Privacy Budgets

We first evaluated whether SFL-PDT provides a more favorable privacy–utility trade-off than fixed-noise private federated learning. The target privacy budget setting was varied as $\epsilon \in \{1, 2, 5\}$, and the reconstruction accuracy of an attacker observing transmitted model updates was recorded. Following Section 3.2.1, these values are treated as target settings for empirical privacy–utility evaluation under the adopted schedule. Lower reconstruction accuracy indicates lower privacy leakage risk. The experimental results are summarized in Table 3.

Table 3. Comparison of reconstruction accuracy under different target privacy budget settings.

Algorithm	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 5$
FedAvg+DP	22.1	30.4	41.2
SFL-PDT	15.9	22.7	34.5

As expected, reconstruction becomes easier as the target privacy budget setting increases, reflecting the standard privacy–utility trade-off. Across all tested settings, however, SFL-PDT consistently yields lower reconstruction accuracy than FedAvg+DP. This result suggests that the terminal-side perturbation and adaptive noise design in SFL-PDT suppress update-level leakage more effectively than fixed-noise perturbation under the same target setting. Meanwhile, the gap remains moderate rather than extreme, which is more consistent with realistic behavior in privacy-preserving federated learning.

4.2.2. Resistance to Gradient Inversion Attacks

We next examined resistance to gradient inversion attacks on the fault diagnosis task. The cosine similarity between attacker-reconstructed parameters and the true parameters was used as a leakage indicator, where a lower value indicates less recoverable information. The results are shown in Figure 3.

The results show that SFL-PDT substantially reduces the leakage similarity relative to both non-private and fixed-noise private baselines. In particular, the reduction from 0.73 to 0.39 compared with FedAvg indicates that terminal-side perturbation significantly weakens the signal available to gradient-based reconstruction. Compared with FedAvg+DP, the remaining improvement suggests that adaptive perturbation is more effective than fixed perturbation in balancing privacy protection and downstream utility. Nevertheless, these results should be interpreted within the evaluated attack setting and do not imply immunity to all possible inversion strategies.

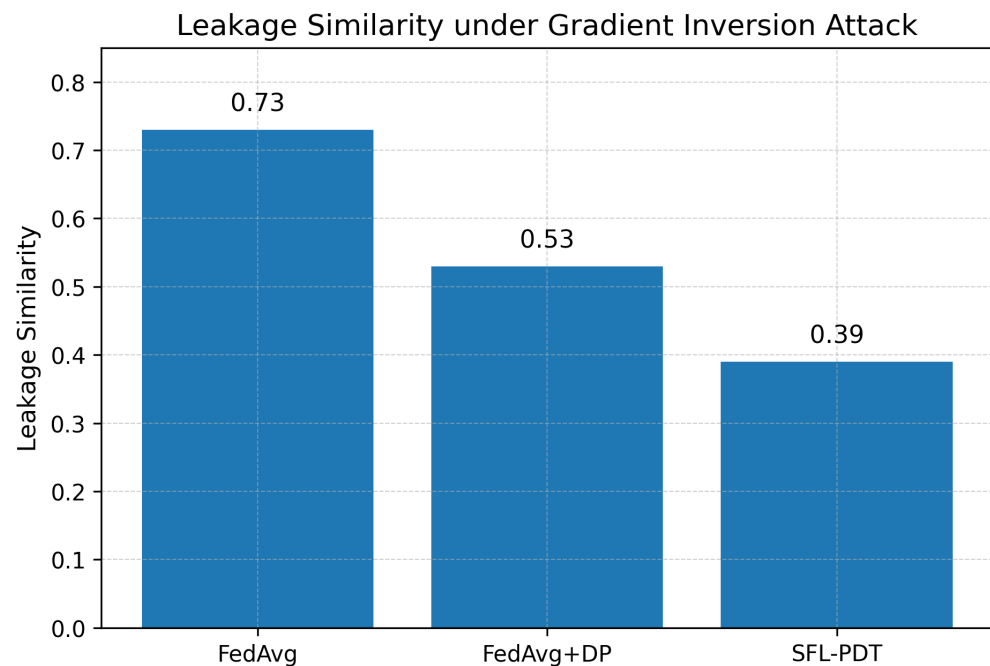


Figure 3. Leakage similarity under gradient inversion attack.

4.2.3. Robustness Under Malicious Client Participation

To evaluate robustness against poisoned client updates, 10% of terminals were set as malicious participants, and their uploaded model updates were corrupted before edge-side regional aggregation. Krum is included as a dedicated Byzantine-robust baseline in this experiment and is implemented at the edge nodes over the received terminal updates. We report the test accuracy under both clean and attacked conditions in Table 4.

Table 4. Comparison of model accuracy under 10% malicious client participation at the edge-side aggregation stage.

Algorithm	Clean Accuracy (%)	Attacked Accuracy (%)	Drop
FedAvg	86.1	75.0	11.1
FedAvg+DP	84.9	77.6	7.3
Krum	85.9	79.7	6.2
SFL-PDT	86.2	80.0	6.2

All methods suffer performance degradation once malicious clients are introduced. Compared with FedAvg and FedAvg+DP, both Krum and SFL-PDT provide clear protection at the edge-side aggregation stage. Krum slightly reduces clean accuracy because of its hard robust selection, whereas SFL-PDT preserves a similar attacked accuracy while maintaining a better clean result. This indicates that the proposed edge-side screening reaches a robustness level close to Krum while remaining better matched to the hierarchical training pipeline.

4.3. Learning Performance Under Heterogeneous Data

4.3.1. Convergence Behavior

We further evaluated whether SFL-PDT improves convergence under heterogeneous client distributions. Figure 4 compares the loss curves of different methods on the Non-IID load forecasting task. A target loss threshold of 0.025 was adopted for convergence comparison.

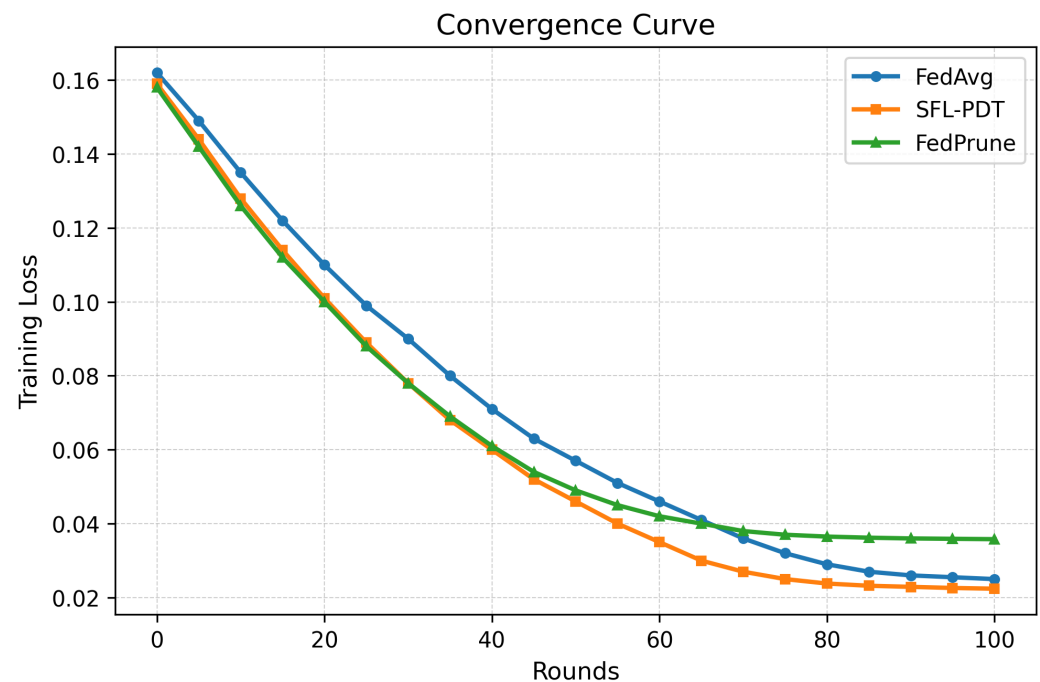


Figure 4. Convergence curve on the Non-IID load forecasting dataset.

Under this criterion, FedAvg requires approximately 100 rounds to reach the target threshold, while SFL-PDT reaches a lower loss level earlier and stabilizes near 0.0224 at the end of training. FedPrune accelerates the early optimization stage due to model reduction, but its loss plateaus at a higher level, remaining around 0.0358 after 100 rounds. These results suggest that the gain of SFL-PDT does not come solely from model compression. Although FedPrune benefits from a smaller effective model size, its later-stage optimization is less stable and converges to a worse solution. By contrast, the similarity-aware aggregation mechanism in SFL-PDT contributes to both faster descent and a lower final loss region, indicating that heterogeneity-aware reweighting is important for stabilizing federated optimization under Non-IID conditions.

4.3.2. Generalization Performance

We next report test-set generalization results on both datasets. The quantitative comparison is given in Table 5.

Table 5. Comparison of generalization performance under heterogeneous regional data.

Algorithm	Load Forecasting MAE	Fault Diagnosis Accuracy (%)
FedAvg	0.024	86.1
FedAvg+DP	0.027	84.9
FedPrune	0.022	87.1
Krum	0.023	87.4
SFL-PDT	0.021	88.2

SFL-PDT achieves the best performance on both tasks, but the margin is moderate. In the regression task, the MAE improvement over FedAvg is noticeable but limited, while FedPrune remains highly competitive. Krum also remains competitive, but its advantage is more limited in this clean heterogeneous setting because it mainly targets abnormal terminal updates at the edge nodes and does not explicitly address cross-edge Non-IID aggregation. The moderate gain of SFL-PDT over Krum is therefore mainly associated

with the central similarity-aware aggregation and the overall hierarchical design. Still, the current evidence is limited to two public datasets and should not be interpreted as universal superiority across all PIoT learning tasks.

4.4. Deployment Efficiency Analysis

4.4.1. Terminal-Side Computation Overhead

Since power data terminals are resource-constrained devices, terminal-side local training cost is a key practical factor. We therefore compared the average local training time per round on the terminal side. The results are shown in Table 6.

Table 6. Comparison of terminal-side local training time.

Algorithm	Local Training Time (s)
FedAvg	4.3
FedAvg+DP	4.6
FedPrune	3.3
SFL-PDT	2.9

SFL-PDT reduces local training time relative to both FedAvg and FedAvg+DP, mainly due to the effective reduction in trainable parameter volume introduced by terminal-oriented compression. At the same time, SFL-PDT remains slightly slower than a pure compression-only lower bound would be, because privacy perturbation and structured update processing still introduce non-negligible overhead. This result indicates that the deployment gain is significant while still reflecting the additional cost of privacy and robustness processing.

4.4.2. Communication Overhead

We next evaluated uplink communication cost by measuring the average payload of a single-round client update. The results are presented in Figure 5.

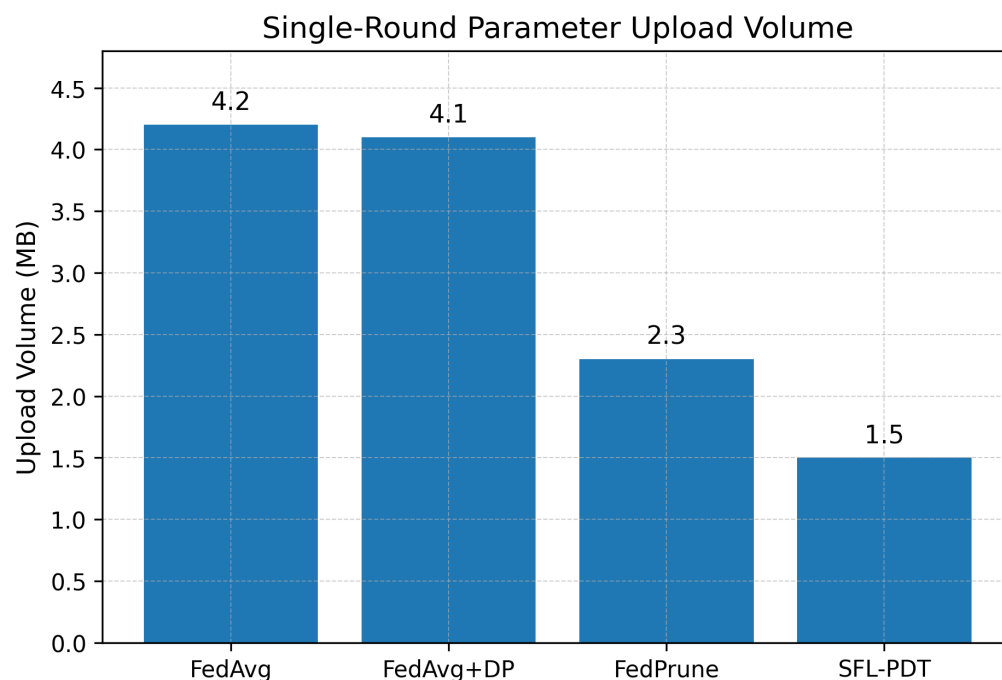


Figure 5. Single-round parameter upload volume.

The proposed method substantially reduces communication volume, which is particularly relevant for wide-area power terminal deployments where uplink bandwidth is often the primary bottleneck. The gain mainly comes from gradient sparsification, lightweight encoding, and quantization. Importantly, the communication benefit is achieved without sacrificing predictive performance, suggesting that the compression strategy functions as an efficiency enhancer rather than an aggressive lossy shortcut.

4.5. Ablation Study

To understand the contribution of each major module, we performed an ablation study by removing one component at a time: local differential privacy (LDP), robust suppression, similarity-aware aggregation, and compression. Because these modules are coupled in the final aggregation pipeline, the leave-one-out setting is used to show the role of each module within the complete SFL-PDT system. The results are summarized in Table 7.

Table 7. Ablation study of SFL-PDT.

Variant	MAE	Accuracy (%)	Comm. Volume (MB)	Training Time (s)
Full SFL-PDT	0.021	88.2	1.5	2.9
w/o LDP	0.020	88.5	1.5	2.8
w/o Robust Suppression	0.022	85.1	1.6	2.9
w/o Similarity-aware Aggregation	0.023	86.5	1.5	3.0
w/o Compression	0.021	88.3	4.1	4.1

Removing LDP slightly improves predictive performance, which is expected because privacy protection introduces a utility cost. However, the gain is small, indicating that the privacy overhead of SFL-PDT remains controllable. Removing robust suppression leads to the most visible drop in classification accuracy, confirming that this module is a major contributor to robustness. Removing similarity-aware aggregation degrades both regression and classification performance and also slightly affects efficiency, suggesting that the modules are not completely decoupled in practice. Finally, removing compression leaves predictive metrics nearly unchanged but sharply increases both communication volume and local training time, showing that compression is the primary driver of deployment efficiency rather than predictive improvement.

Overall, the strength of SFL-PDT lies not in any single component, but in the complementary interaction among privacy protection, robustness control, heterogeneity-aware aggregation, and efficiency-oriented compression.

A progressive ablation that adds modules one by one from FedAvg could provide a finer-grained comparison and will be explored in future work.

4.6. Sensitivity Analysis

4.6.1. Sensitivity to Privacy Budget

To further examine the privacy–utility trade-off, we varied the target privacy budget setting from $\epsilon = 0.5$ to $\epsilon = 8$, and jointly recorded reconstruction accuracy and downstream fault diagnosis accuracy. The curve is used to compare the empirical privacy–utility behavior produced by the adopted perturbation schedule. The results are shown in Figure 6.

As the target privacy budget increases, reconstruction accuracy rises for both private methods, while downstream task accuracy also improves gradually. This monotonic trend is consistent with the expected privacy–utility trade-off. Across the full budget range, SFL-PDT consistently maintains lower reconstruction accuracy and higher task accuracy than FedAvg+DP, although the utility gap becomes slightly smaller in the high- ϵ regime. This

suggests that the proposed adaptive perturbation mechanism provides a more favorable operating curve rather than merely a single-point advantage.

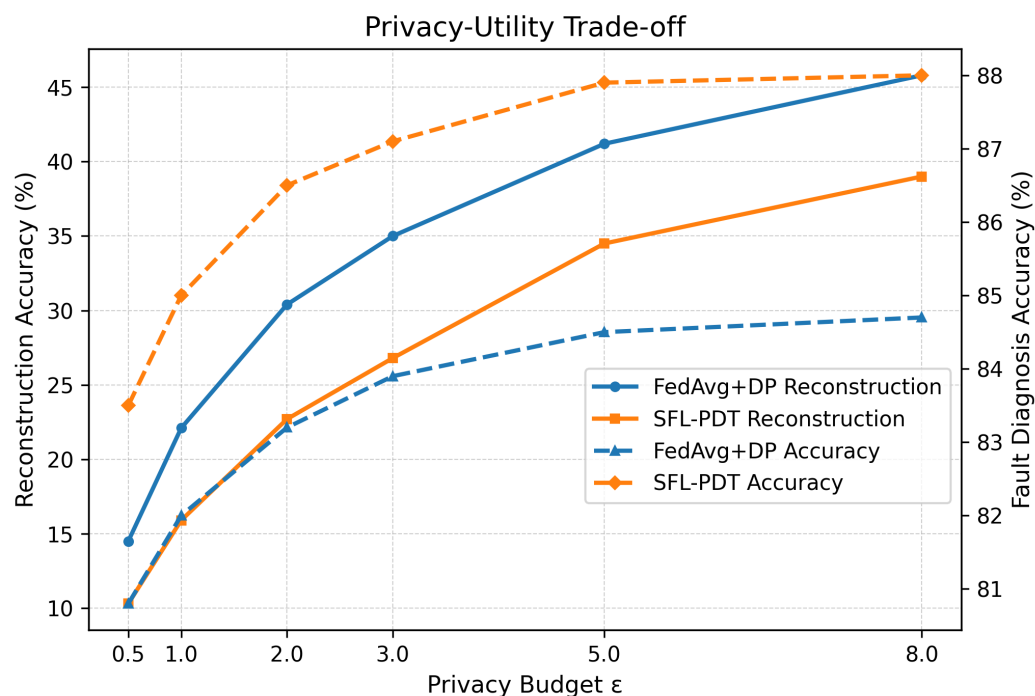


Figure 6. Privacy–utility trade-off under different target privacy budget settings.

4.6.2. Sensitivity to Malicious Client Ratio

We then varied the malicious client ratio from 0% to 30% and measured the corresponding fault diagnosis accuracy. The results are shown in Figure 7.

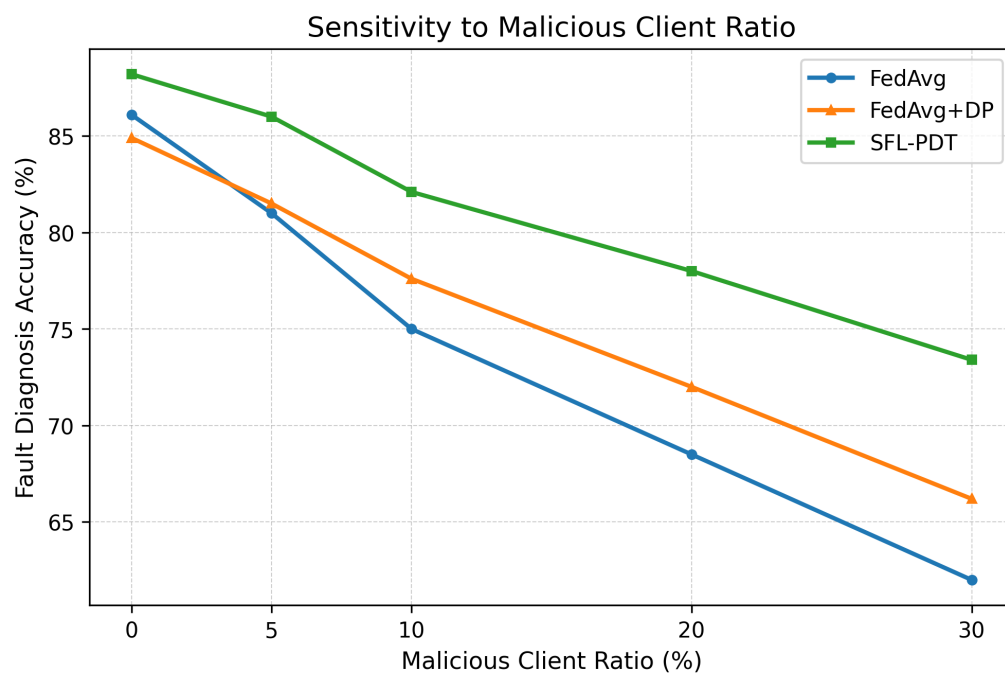


Figure 7. Sensitivity to malicious client ratio.

The accuracy of all methods decreases as the malicious participation ratio increases. SFL-PDT remains consistently above the non-robust baselines across the evaluated range,

although the margin is moderate. This trend suggests that the proposed robust aggregation strategy remains beneficial when the attack intensity increases beyond the default 10% setting.

4.6.3. Sensitivity to Pruning Ratio

Finally, we varied the pruning ratio from 0.0 to 0.5 to study the trade-off between efficiency and predictive performance. The results are shown in Figure 8.

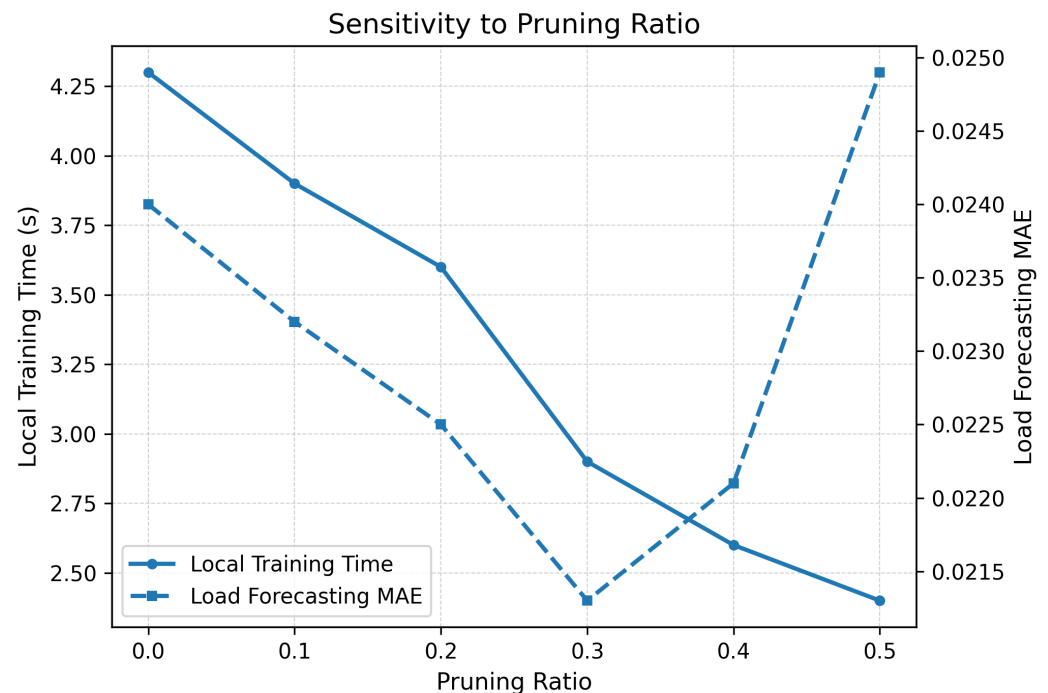


Figure 8. Sensitivity of training efficiency and predictive performance to pruning ratio.

Increasing the pruning ratio consistently reduces local training time, while the forecasting MAE first improves and then deteriorates. This indicates the presence of a practical “sweet spot” in compression aggressiveness: moderate pruning can improve efficiency and mildly regularize the model, whereas excessive pruning harms representational capacity. In our setting, a pruning ratio around 0.3 yields the most balanced efficiency–performance trade-off.

4.7. Discussion

Taken together, the above experiments indicate that SFL-PDT provides a better coordinated balance among privacy protection, robustness, convergence stability, predictive performance, and deployment efficiency than the compared baselines under the evaluated conditions. These gains are not attributable to a single design choice. Rather, the proposed framework benefits from the coordinated combination of terminal-side perturbation, anomaly-aware aggregation, heterogeneity-aware weighting, and terminal-oriented compression.

From a practical perspective, the results are especially relevant for power data terminals because the targeted deployment environment is characterized by three simultaneous constraints: strict data sensitivity, heterogeneous local distributions, and limited communication/computation resources. In the current design, homogeneity is mainly assumed within each edge node, while cross-edge heterogeneity is handled at the central server. In this context, optimizing only one dimension is insufficient. A privacy-only solution may degrade utility, a compression-only solution may fail under heterogeneity, and a robustness-

only solution may not address deployment overhead. The empirical results suggest that jointly designing these components yields a more balanced system-level trade-off.

The comparison with Krum at the edge-side aggregation stage shows that the robustness of SFL-PDT is close to that of a dedicated robust aggregator under malicious client participation. This suggests that the main benefit of SFL-PDT lies in combining edge-side robustness with privacy protection, heterogeneity-aware aggregation, and terminal-side compression within one training pipeline. At the same time, the current design is most suitable for hierarchical settings in which regional grouping at the edge is meaningful. Broader validation under larger-scale deployments, stronger attacks, and stricter privacy accounting remains an important direction for future work.

Beyond the current empirical evaluation, future application-oriented validation should also consider more realistic power-electronics scenarios. Recent studies on fault diagnosis in cascaded H-bridge multilevel converters and stability root-cause identification in solid-state transformers show that practical terminal-side intelligence often requires reliable monitoring, diagnosis, and stability assessment under constrained and heterogeneous operating conditions [41,42]. These tasks are closely aligned with the deployment assumptions of SFL-PDT and can serve as meaningful testbeds for evaluating whether privacy-preserving, robust, and resource-aware federated learning can generalize to broader power-electronic equipment.

5. Conclusions

In this paper, we proposed SFL-PDT, a secure and efficient federated learning framework for power data terminals, to address three practical challenges in PIIoT-oriented collaborative learning: update-level privacy leakage, heterogeneous client data distributions, and limited computation and communication resources at terminal devices. Within this framework, edge nodes aggregate relatively homogeneous terminal updates, while the central server handles aggregation across heterogeneous regions. To this end, terminal-side privacy protection, anomaly-aware robust aggregation, similarity-aware weighted updating, and terminal-oriented compression are integrated into a unified hierarchical federated learning framework.

Experimental results on two representative power-related tasks demonstrate that, under the evaluated settings, SFL-PDT achieves a more favorable overall trade-off than the compared baselines. Specifically, it reduces reconstruction-related leakage and retains stable performance under malicious client participation; it shows faster and more stable convergence under heterogeneous client distributions while maintaining better test performance on both regression and classification tasks; and it substantially lowers terminal-side training cost and uplink communication volume, which is important for deployment in resource-constrained power terminal environments.

A key implication of this study is that federated learning for power data terminals should not be optimized along a single axis only. In such environments, privacy, robustness, heterogeneity adaptation, and efficiency are tightly coupled system requirements. The empirical evidence in this work suggests that coordinating these components within one framework is more effective than applying them independently.

Nevertheless, several limitations remain. First, the current evaluation is based on two public datasets and a controlled hierarchical testbed, and therefore does not yet fully capture the complexity of large-scale industrial PIIoT deployments. Second, the threat model in this work mainly covers update reconstruction and malicious parameter corruption, while stronger adversarial settings, such as collusive attacks and backdoor-oriented attacks, were not investigated. Third, the present privacy accounting and adaptive scheduling strategies are still relatively simplified and would benefit from more rigorous optimization

and theoretical analysis. Thus, the reported privacy budget settings correspond to empirical settings under the adopted perturbation schedule unless a full composition accountant is employed.

Future work will therefore focus on three directions. First, more rigorous privacy accounting and more adaptive privacy budget allocation will be introduced across communication rounds. Second, the framework will be evaluated under broader and stronger adversarial settings, including collusive and backdoor attacks. Third, the proposed design will be validated in more realistic large-scale PLoT environments with dynamic client participation and time-varying network conditions, thereby further improving the practical relevance of secure federated learning for power systems.

Author Contributions: Conceptualization, X.D. and X.G.; methodology, X.D. and C.L.; software, C.L. and J.H.; validation, J.H., Y.L. and R.Y.; formal analysis, X.D. and C.L.; investigation, Y.L. and R.Y.; resources, X.G. and R.Y.; data curation, J.H. and Y.L.; writing—original draft preparation, X.D. and C.L.; writing—review and editing, X.G., R.Y. and Y.Z.; visualization, J.H. and Y.L.; supervision, X.G.; project administration, X.G. and R.Y.; funding acquisition, X.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the science and technology project of Beijing Electric Power Company under grant “Research on Distributed Business Security Protection Technology for New Power System” (520230250002).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors X.D., C.L., J.H. and Y.L. were employed by State Grid Beijing Electric Power Co., Ltd. Authors X.G., R.Y. and Y.Z. were employed by China Electric Power Research Institute. All authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The authors declare that this study received funding from the Science and Technology Project of Beijing Electric Power Company under grant “Research on Distributed Business Security Protection Technology for New Power System” (520230250002). The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

References

1. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Aguera y Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017*; PMLR: Fort Lauderdale, FL, USA, 2017; pp. 1273–1282.
2. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19.
3. Fredrikson, M.; Jha, S.; Ristenpart, T. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS), Denver, CO, USA, 12–16 October 2015*; ACM: New York, NY, USA, 2015; pp. 1322–1333.
4. Zhu, L.; Liu, Z.; Han, S. Deep Leakage from Gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Vancouver, BC, Canada, 2019; Volume 32.
5. Han, S.; Mao, H.; Dally, W.J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In *Proceedings of the 4th International Conference on Learning Representations (ICLR), San Juan, PR, USA, 2–4 May 2016*; ICLR: San Juan, PR, USA, 2016.
6. Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; Vojnovic, M. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Long Beach, CA, USA, 2017; Volume 30.
7. Liu, L.; Zhang, J.; Song S.; Letaief, K.B. Client-Edge-Cloud Hierarchical Federated Learning. In *Proceedings of the 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020*; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.

8. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated Optimization in Heterogeneous Networks. In *Proceedings of the Machine Learning and Systems (MLSys)*, Austin, TX, USA, 2–4 March 2020; MLSys: Austin, TX, USA, 2020; Volume 2, pp. 429–450.
9. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Robust and Communication-Efficient Federated Learning from Non-IID Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3400–3413. [[PubMed](#)]
10. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [[CrossRef](#)]
11. Saputra, Y.M.; Hoang, D.T.; Nguyen, D.N.; Dutkiewicz, E.; Mueck, M.D.; Srikathyayani, S. Energy Demand Prediction with Federated Learning for Residential Grid Systems. In *Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, HI, USA, 9–13 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
12. Zhang, W.; Li, X.; Ma, H.; Luo, Z.; Li, Z. Federated Learning for Machinery Fault Diagnosis with Dynamic Validation and Self-Supervision. *Knowl. Based Syst.* **2021**, *213*, 106679. [[CrossRef](#)]
13. Su, Z.; Wang, Y.; Xu, Q.; Fei, M.; Tian, Y.-C.; Zhang, N. A Secure Charging Scheme for Electric Vehicles with Smart Communities in Energy Blockchain. *IEEE Internet Things J.* **2019**, *6*, 4601–4614.
14. Liu, H.; Zhang, X.; Shen, X.; Sun, H. A Federated Learning Framework for Smart Grids: Securing Power Traces in Collaborative Learning. *arXiv* **2021**, arXiv:2103.11870.
15. Geiping, J.; Bauermeister, H.; Dröge, H.; Moeller, M. Inverting Gradients—How Easy Is It to Break Privacy in Federated Learning? In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Vancouver, BC, Canada, 2020; Volume 33, pp. 16937–16947.
16. Zhao, B.; Mopuri, K.R.; Bilen, H. iDLG: Improved Deep Leakage from Gradients. *arXiv* **2020**, arXiv:2001.02610.
17. Reisizadeh, A.; Mokhtari, A.; Hassani, H.; Jadbabaie, A.; Pedarsani, R. FedPAQ: A Communication-Efficient Federated Learning Method with Periodic Averaging and Quantization. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*; PMLR: Palermo, Italy, 2020; Volume 108, pp. 2021–2031.
18. Diao, E.; Ding, J.; Tarokh, V. HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients. In *International Conference on Learning Representations (ICLR)*; OpenReview.net: Amherst, MA, USA, 2021.
19. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.J.; Stich, S.U.; Suresh, A.T. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*; PMLR: Cambridge, MA, USA, 2020; Volume 119, pp. 5132–5143.
20. Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; Poor, H.V. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Vancouver, BC, Canada, 2020; Volume 33, pp. 7611–7623.
21. Acar, D.A.E.; Zhao, Y.; Navarro, R.M.; Mattina, M.; Whatmough, P.N.; Saligrama, V. Federated Learning Based on Dynamic Regularization. In *International Conference on Learning Representations (ICLR)*; OpenReview.net: Amherst, MA, USA, 2021.
22. Li, X.; Jiang, M.; Zhang, X.; Kamp, M.; Dou, Q. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. In *International Conference on Learning Representations (ICLR)*; OpenReview.net: Amherst, MA, USA, 2021.
23. Li, Q.; He, B.; Song, D. Model-Contrastive Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Nashville, TN, USA, 2021; pp. 10713–10722.
24. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, Dallas, TX, USA, 30 October–3 November 2017; ACM: New York, NY, USA, 2017; pp. 1175–1191.
25. Geyer, R.C.; Klein, T.; Nabi, M. Differentially Private Federated Learning: A Client Level Perspective. *arXiv* **2017**, arXiv:1712.07557.
26. Pillutla, K.; Kakade, S.M.; Harchaoui, Z. Robust Aggregation for Federated Learning. *IEEE Trans. Signal Process.* **2022**, *70*, 1142–1154. [[CrossRef](#)]
27. Xie, C.; Chen, M.; Chen, P.Y.; Li, B. CRFL: Certifiably Robust Federated Learning against Backdoor Attacks. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*; PMLR: Cambridge, MA, USA, 2021; Volume 139, pp. 11372–11382.
28. Dwork, C. Differential Privacy: A Survey of Results. In *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation (TAMC)*, Xi'an, China, 25–29 April 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1–19.
29. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, Y.; Talwar, K.; Zhang, L. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, Vienna, Austria, 24–28 October 2016; ACM: New York, NY, USA, 2016; pp. 308–318.
30. Noble, M.; Bellet, A.; Dieuleveut, A. Differentially Private Federated Learning on Heterogeneous Data. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*; PMLR: Cambridge, MA, USA, 2022; Volume 151, pp. 1397–1412.

31. Blanchard, P.; El Mhamdi, E.M.; Guerraoui, R.; Stainer, J. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Long Beach, CA, USA, 2017; Volume 30, pp. 119–129.
32. Yin, D.; Chen, Y.; Kannan, R.; Bartlett, P. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*; PMLR: Stockholm, Sweden, 2018; Volume 80, pp. 5650–5659.
33. Xie, C.; Koyejo, O.; Gupta, I. Zeno: Distributed Stochastic Gradient Descent with Suspicion-Based Fault-Tolerance. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*; PMLR: Long Beach, CA, USA, 2019; Volume 97, pp. 6893–6901.
34. Fung, C.; Yoon, C.J.M.; Beschastnikh, I. Mitigating Sybils in Federated Learning Poisoning. *arXiv* **2018**, arXiv:1808.04866.
35. Reddi, S.J.; Charles, Z.; Zaheer, M.; Garrett, Z.; Rush, K.; Konečný, J.; Kumar, S.; McMahan, H.B. Adaptive Federated Optimization. In *International Conference on Learning Representations (ICLR)*; OpenReview.net: Amherst, MA, USA, 2021.
36. Karimireddy, S.P.; Jaggi, M.; Kale, S.; Mohri, M.; Reddi, S.J.; Stich, S.U.; Suresh, A.T. Mime: Mimicking Centralized Stochastic Algorithms in Federated Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 18870–18883.
37. Li, Z.; Chaturvedi, P.; He, S.; Chen, H.; Singh, G.; Kindratenko, V.V.; Huerta, E.A.; Kim, K.; Madduri, R.K. FedCompass: Efficient Cross-Silo Federated Learning on Heterogeneous Client Devices Using a Computing Power-Aware Scheduler. In *International Conference on Learning Representations (ICLR)*; OpenReview.net: Amherst, MA, USA, 2024.
38. Bottou, L.; Bousquet, O. The Tradeoffs of Large Scale Learning. In *Advances in Neural Information Processing Systems (NIPS)*; MIT Press: Vancouver, BC, Canada, 2008; Volume 20, pp. 161–168.
39. PJM Interconnection. Public Electric Load Dataset. US PJM Power Market, 2018–2022. Available online: <https://www.pjm.com/> (accessed on 15 December 2025).
40. IEEE PHM Society. IEEE PHM 2012 Power Transformer Fault Diagnosis Dataset. In *Proceedings of the 2012 IEEE Prognostics and Health Management Conference*; IEEE: Denver, CO, USA, 2012; pp. 1–8.
41. Lin, H.; Chung, H.S.-H.; Lin, C.; Xie, D.; Deng, Q.; Lyu, M.; Goetz, S.M.; Chen, J.; Ge, X. Improved Fault Diagnosis Capability in CHBMCs: Counter Design for Multiple OC Switches via an E-SVM Unit. *IEEE Trans. Power Electron.* **2026**, *41*, 2358–2376. [[CrossRef](#)]
42. Meng, X.; Xie, D.; Lin, H.; Lin, C.; Ge, X.; Liu, Z. Dissipativity-Based Multiport Stability Root-Cause Identification and Mitigation for Solid-State Transformers. *IEEE Trans. Ind. Electron.* **2026**, *73*, 1–13. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.