C-PFL: A committee-based personalized federated learning framework

Lifan Pan, Hao Guo, Wanxin Li

PII: S1084-8045(25)00224-3
DOI: https://doi.org/10.1016/j.jnca.2025.104327
Reference: YJNCA 104327

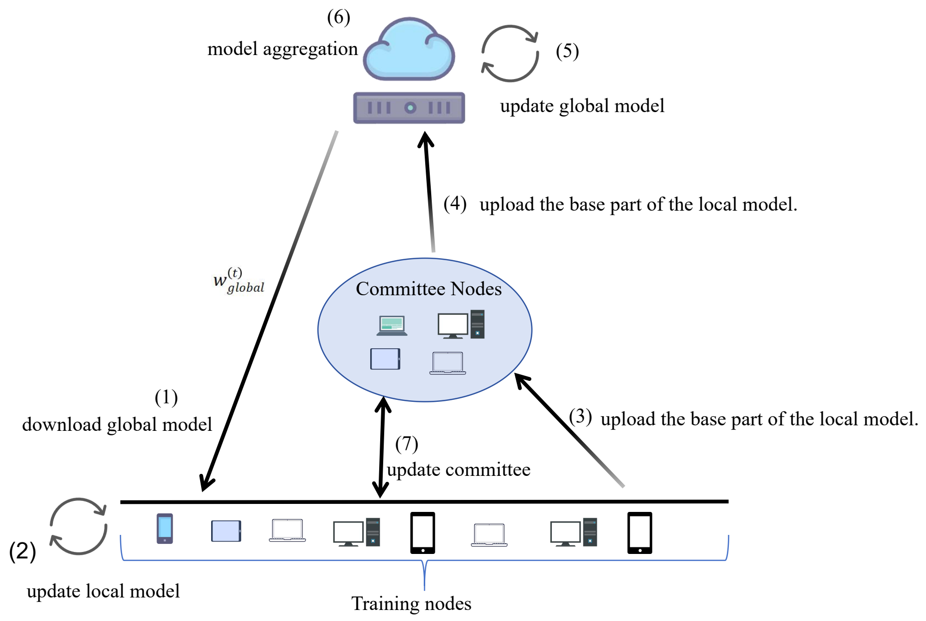
To appear in: *Journal of Network and Computer Applications*

Received date : 9 May 2025
Revised date : 20 August 2025
Accepted date : 9 September 2025

Graphical Abstract

C-PFL: A Committee-Based Personalized Federated Learning Framework

Lifan Pan, Hao Guo*, Wanxin Li*



This paper proposes a novel personalized federated learning framework called C-PFL, which enhances robustness and personalization in federated learning under non-IID data and adversarial client settings. C-PFL separates the model into a shared base component and a personalized head component, where only the base is updated during federated training. To ensure the quality and reliability of model aggregation, C-PFL introduces a committee-based validation mechanism, where a selected group of high-contribution clients verifies submitted model updates. This process effectively filters out low-quality or malicious contributions, stabilizing and securing the global model.

Highlights

C-PFL: A Committee-Based Personalized Federated Learning Framework

Lifan Pan, Hao Guo*, Wanxin Li*

- We propose C-PFL, a committee-based personalized federated learning framework.
- C-PFL filters low-quality updates using peer validation without relying on public data.
- The model splits into a shared backbone and a client-specific head for personalization.
- A contribution-based dynamic committee mechanism ensures robustness against attacks.
- C-PFL outperforms six baselines in accuracy by up to 2.89% in benign settings and up to 6.96% under 40% malicious clients.

C-PFL: A Committee-Based Personalized Federated Learning Framework

Lifan Pan[a], Hao Guo*[a], Wanxin Li*[b]

[a] *School of Software, Northwestern Polytechnical University, Xi'an, 710129, China*
[b] *School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, 215123, China*

Abstract

Federated Learning (FL) is an emerging machine learning paradigm that enables multiple parties to train a shared model while preserving data privacy collaboratively. However, malicious clients pose a significant threat to FL systems. This interference not only deteriorates model performance but also exacerbates the unfairness of the global model caused by data heterogeneity, leading to inconsistent performance across clients. We propose C-PFL, a committee-based personalized FL framework that improves both robustness and personalization. In contrast to prior approaches such as FedProto (which relies on the exchange of class prototypes), Ditto (which employs regularization between global and local models), and FedBABU (which freezes the classifier head during federated training), C-PFL introduces two principal innovations. C-PFL adopts a split-model design, updating only a shared backbone during global training while fine-tuning a personalized head locally. A dynamic committee of high-contribution clients validates submitted updates without public data, filtering low-quality or adversarial contributions before aggregation. Experiments on MNIST, Fashion-MNIST, CIFAR-10, CIFAR-100, and AGNews show that C-PFL outperforms six state-of-the-art personalized FL baselines by up to 2.89% in non-adversarial settings, and by as much as 6.96% under 40% malicious clients. These results demonstrate C-PFL's ability to sustain high accuracy and stability across diverse non-IID scenarios, even with significant adversarial participation.

Keywords:
Federated learning, Personalization, Model validation, Aggregation

1. Introduction

Federated learning (FL) is a distributed machine learning paradigm introduced by Google in 2017[1], designed to enable multiple participants to collaboratively train a shared model without transferring their raw data to a central server. Instead, model training is performed locally on each client device, and only model updates are exchanged, thereby preserving data privacy and reducing regulatory concerns associated with sensitive information. This privacy-by-design property makes FL particularly attractive in domains such as healthcare, where medical records cannot be shared due to confidentiality requirements; finance, where transaction histories are highly sensitive; and the Internet of Things (IoT)[2], where numerous edge devices generate privacy-critical data. Beyond privacy, FL offers the ability to leverage diverse and geographically distributed datasets, potentially improving model generalization across heterogeneous environments.

The core idea of Federated Learning (FL) is to retain data locally on client devices instead of uploading it to a centralized server, with Federated Averaging (FedAvg) [3] being a commonly used aggregation method. Figure 1 illustrates the overall FL architecture. 1) The typical FL training process begins with Model Initialization, where the central server initializes and distributes the global model to all participating clients. 2) This is followed by Local Training, during which each client trains the model exclusively on its local device data without sharing the raw data. 3) Subsequently, clients perform Model Update Uploading, returning only their locally updated model parameters to the server. 4) The server then conducts Global Aggregation, typically using weighted averaging, to integrate these updates into a new global model. This entire sequence forms an Iterative Process: the updated global model is redistributed to clients for further local training, and the cycle repeats until model convergence is achieved. This paradigm enables collaborative learning that preserves privacy over distributed data.

Despite its advantages, FL faces several challenges. First, data across clients is often non-identically and independently distributed (non-IID), which can lead to significant variations in local model updates, thus impairing the convergence and stability of the global model. Additionally, balancing global generalization and local personalization remains a core challenge. Due to the diverse characteristics of the data and the task requirements between clients, a global model that is one size fits all often does not meet personalized needs [4, 5].

Moreover, the decentralized nature of FL, where only model updates are shared (not raw data), inherently exposes the system to a spectrum of adversarial threats orchestrated by malicious clients. These adversaries can intentionally manipulate the training process to compromise the global model's integrity through various means:

- Data/Model Poisoning Attacks: Malicious clients can manipulate their local training data or craft malicious model updates to introduce biases, corrupt the learning objective, or degrade the global model's performance on specific tasks or inputs. This work primarily addresses input data poisoning attacks, aiming to detect and mitigate their impact during the federated aggregation process.

- Backdoor Attacks: Attackers can embed hidden triggers into their local data or model updates. The resulting global model behaves normally on clean inputs but misclassifies inputs containing the specific trigger pattern.

- Byzantine Attacks: Malicious clients (acting as "Byzantine" nodes) can arbitrarily deviate from the protocol. This includes sending fabricated data distributions, arbitrarily corrupted model updates (e.g., random noise, sign-flipped gradients), or selectively dropping communications to disrupt convergence or steer the model toward a malicious state.

This paper focuses on the prevalent and critical challenge of poisoning attacks in the context of FL. These attacks pose significant risks to the reliability and trustworthiness of the federated learning process, and robust defenses are essential.

To solve the problem that the unified global model cannot meet the personalized needs of each node due to data heterogeneity in federated learning, this paper proposes a committee-based personalized federated learning algorithm (C-PFL). The algorithm aims to achieve model personalization and defend against poisoning attacks. C-PFL introduces a committee mechanism, in which the committee node verifies the model updates submitted by other nodes in each training round (without the need for a public dataset), ensuring that only qualified updates participate in global aggregation. This enables most honest nodes to enhance each other and jointly improve the global model, while ignoring a small number of incorrect or malicious updates, thereby maintaining high stability under malicious attacks and only adding a small amount of verification overhead. To better adapt to the unique data distribution and task requirements of each node, C-PFL divides the model into a public part (base) and a personalized part (head). During the training process, nodes only upload the base part for global aggregation; after obtaining the global public model, each node fine-tunes the head part locally, ultimately achieving a highly personalized model.

In summary, our main contributions are as follows:

- **Committee-based robust personalized FL framework**: We propose C-PFL, which integrates a split-model design with a dynamic committee mechanism to enhance both robustness and personalization in federated learning. The backbone is trained globally while the head is personalized locally.
- **Dynamic committee selection and thresholding**: A contribution-based selection strategy forms the committee, and a dynamically updated accuracy threshold filters low-quality or malicious updates without requiring public datasets.
- **Comprehensive evaluation under non-IID and adversarial settings**: Extensive experiments on five benchmark datasets demonstrate that C-PFL outperforms six state-of-the-art personalized FL methods, achieving up to 2.89% higher accuracy in benign conditions and up to 6.96% under 40% malicious clients.

2. Related work

FedAvg [3] is one of the most widely adopted algorithms in federated learning. It aggregates model updates from all clients using weighted averaging, typically based on the number of local samples, to generate a new global model. However, in real-world scenarios, the data across clients is often heterogeneous (non-IID). In such cases, the performance of FedAvg tends to degrade significantly due to the divergent directions of local updates, leading to unstable global model training [6]. As a result, the global model often fails to generalize well to clients whose local data distributions differ substantially from the overall average. For practical applications that frequently deal with non-IID local datasets, relying on a single global model is usually insufficient [7].

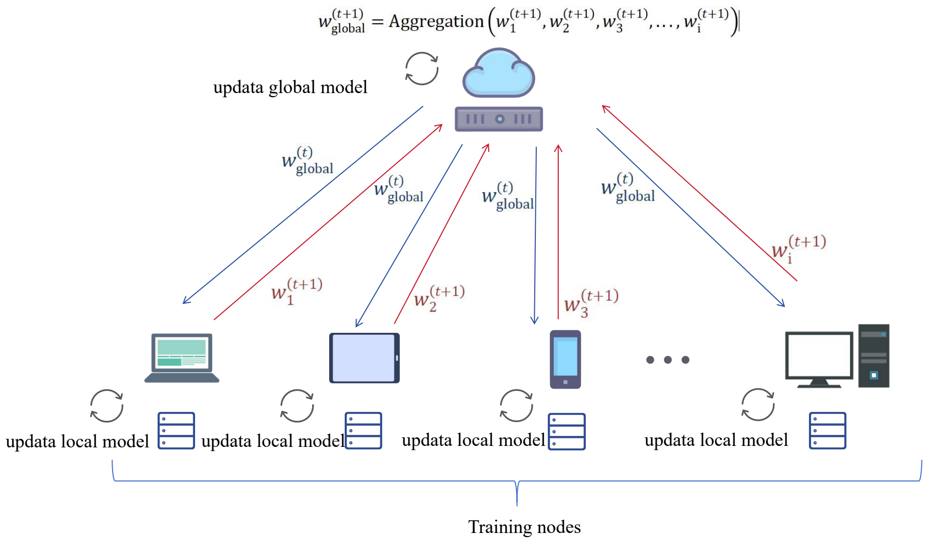


Figure 1: Framework of traditional federated learning.

In recent years, personalization has gained significant attention to address statistical heterogeneity and enable customized models in federated learning. Existing personalized federated learning (PFL) methods can be broadly categorized into the following four types: 1) local fine-tuning-based personalization, 2) knowledge distillation-based personalization, 3) model splitting-based personalization, and 4) regularization-based personalization.

2.1. Local fine-tuning-based personalization

Through differentiated training strategies, each client can obtain a model that better fits its data characteristics, thereby improving personalization and avoiding the generalization drop caused by data heterogeneity. FedPHP [8] emphasizes the importance of historical personalized models. It proposes the *Inherited Private Model* (HPM), which temporally ensembles each client's historical personalized models and uses them to supervise the next round of global personalization. APFL [9] presents an adaptive personalized federated learning algorithm that derives generalization bounds for the mixture of local and global models and learns the optimal mixing parameter. Thus, each client contributes to the global model while training its local model. FedSAE [10] is an adaptive FL system that accelerates the convergence of the global model by adaptively selecting clients with higher local training loss in each round. To address the challenges of statistical heterogeneity, FedALA [11] supports client models in personalized FL by capturing informative representations required by the global model.

FedCA-TDD [12] modifies the aggregation strategy of the global model: the feature extraction layer is still weighted by the number of local samples, while the classification layer is aggregated based on the contribution of each class.

APPLE [13], *FedFomo* [14], and *FedAMP* [15] enable model personalization through client-to-client collaboration. APPLE adaptively adjusts the degree of model sharing between clients and dynamically balances the focus between global and local objectives. It learns how much each client can benefit from others, thereby mitigating the challenges posed by non-IID datasets. FedFomo computes an optimal weighted combination of peer models for each client, allowing them to collaborate only with relevant peers. This strategy optimizes models individually rather than relying on a single global average, offering greater personalization flexibility. FedAMP promotes peer-to-peer collaboration among clients with similar data distributions through message passing, enhancing cooperative training among similar clients.

FedSAE [10] and *TiFL* [16] improve the global model via intelligent client selection. FedSAE adaptively selects clients with high local loss for each training round, thus accelerating convergence. TiFL organizes clients into tiers based on training performance and adaptively selects participating clients from the same tier to optimize accuracy and training efficiency.

2.2. Knowledge distillation-based federated learning

These methods leverage global semantic knowledge to explicitly align local and global feature representations, resulting in improved representations. Additionally, they quantify the benefits of combining classifiers from different clients and formulate an optimization problem to estimate the optimal weights. This approach enhances generalization through global knowledge and enables effective collaboration among local classification heads.

FedDistill [17] introduces two algorithms—Federated Distillation (FD) and Federated Augmentation (FAug). FD significantly reduces communication overhead compared to traditional FL, especially for large models. FAug complements this by jointly training a generative model to augment local data. PCTPF [18] integrates knowledge distillation with model fusion in a two-stage training process. In the first stage, the server aggregates client knowledge into a teacher model via regularized KD and distills it into a student model for local training to alleviate client drift. In the second stage, clients employ an adaptive weighting mechanism to fuse the global and local models, yielding personalized models for improved performance. FedProto [19] enables clients and server communication via abstract class prototypes rather than gradients. These prototypes are aggregated on the server and sent back to clients to guide local training. FedPAC [20] enhances model personalization by aligning local and global features and promoting cooperation among classification heads. It improves representation learning by incorporating global knowledge and optimizing the combination of classifiers for each client.

2.3. Model splitting-based federated learning

This approach partitions the model into multiple components, allowing each client to train only the parts relevant to specific data features or input variables, or to handle certain elements in a specialized manner. A series of methods—LG-FedAvg [21], FedGH [22], GPFL [23], and FedRep [24]—highlight the importance of both global and local information, aiming to train a personalized model alongside a shared global model.

LG-FedAvg reduces communication overhead and improves training scalability and efficiency by jointly learning compact local representations and a global model. FedGH and FedRep recognize that although data in federated settings is often non-IID, clients usually share a common global feature representation. At the same time, statistical heterogeneity primarily exists in the label space. These methods leverage local data representations submitted by clients to jointly train a generalized global prediction head and personalized local heads. GPFL focuses on feature extraction in personalized FL by simultaneously learning global and personalized features on each client.

FedBABU [6] and FedPer [25] specifically focus on personalized models by decomposing the entire network into a backbone (extractor) responsible for general representation learning and a head (classifier) tailored for personalization. In FedPer, the backbone and head are trained during federated learning. In contrast, FedBABU only updates the backbone in the federated phase and fine-tunes the head during evaluation to achieve personalization.

FedGC [26] addresses privacy concerns in face recognition tasks. Traditional decentralized FL methods share all model parameters across clients, which may lead to privacy leakage in facial recognition scenarios. FedGC introduces a softmax-based regularizer that injects inter-client gradient terms to correct class embedding gradients. This correction is applied from a backpropagation perspective to enhance privacy preservation. FedGen [27] proposes a data-free distillation framework that transfers knowledge to FL clients. A generative model is trained on the FL server and distributed to clients. Each client uses the distilled knowledge as an inductive bias to generate enhanced representations in the feature space, thereby improving local learning performance.

2.4. Regularization-based federated learning

Model regularization is a widely adopted strategy in machine learning to prevent overfitting and improve convergence. In federated learning (FL), regularization can be employed to constrain the influence of local updates, thereby enhancing the stability and generalizability of the global model. This, in turn, facilitates the development of better personalized models [7].

pFedMe [28] addresses the limitations in global model performance arising from statistical heterogeneity across clients by employing Moreau Envelopes as a regularized local loss function. MOON [29] aims to minimize the divergence of representations between local and global models, effectively reducing the divergence of weights, while simultaneously maximizing the distance between current and previous local model representations to accelerate convergence. This novel approach enables clients to learn representations closer to the global model, thereby mitigating local model inconsistency.

FedProx [30] and Ditto [31] improve FL performance by introducing explicit regularization between global and local objectives. FedProx adds a proximal term to the local objective to ensure client updates do not drift too far from the global model. This additional constraint improves convergence under non-IID data distributions. Ditto introduces a regularization strategy to bridge the gap between global and personalized models, aiming to provide better fairness and robustness, especially for underrepresented or highly heterogeneous clients.

Our proposed *C-PFL* belongs to model splitting-based federated learning. Designed for federated systems with potentially malicious or unreliable participants, C-PFL integrates a committee-based mechanism that limits the impact of such participants on the global model, thereby enhancing robustness against adversarial threats. Moreover, each client must maintain consistency only in the base part of the model, while being free to customize its head, allowing for effective personalization. A comparative summary of the optimization strategies used in various federated learning algorithms is presented in Table 1.

To provide a clearer understanding of how our proposed C-PFL differs from existing federated learning methods, we summarize and compare the key characteristics of FedAvg, FedGH, FedProto, and C-PFL in terms of model architecture, training strategy, and aggregation mechanism in Table 2. While FedAvg adopts a monolithic global model and simple averaging, and FedProto relies on prototype transmission, our method presents a novel committee-based validation mechanism and a split model structure, offering better personalization and robustness.

2.5. Other Advances in Federated Learning

Recent developments in federated learning have introduced novel directions that complement robustness and personalization, including zero-trust security modeling, transformer-based domain adaptation, structured generalization, and data-free robustness enhancement.

FedKCSS[32] proposed a zero-trust FL framework with dynamic trust evaluation and knowledge transfer, which complements our committee-based validation as a performance-driven trust mechanism. FedDAvT[33] applied transformer-based domain adaptation for Alzheimer's diagnosis, demonstrating the potential of attention mechanisms in FL.

FedPartWhole[34] improves domain generalization through consistent part-whole hierarchies, emphasizing structural invariance across clients. DFRD[35], a data-free robustness distillation method that aligns with our goal of defending against adversarial clients and could enhance committee evaluation in future extensions.

DGFedRS[36] employs diffusion augmentation and guided denoising to enhance the diversity of the latent data distribution, suppress noise, and retain user-specific preferences in sequential recommendation tasks. FedRL[37] introduces a reinforcement learning-based framework with a Reinforcement Selector for adaptive client participation and a Hypernet Generator to reduce communication costs, enabling efficient deployment and model updates. PCFedRec[38] captures fine-grained heterogeneous dependencies in multi-behavior consumer data via a Fine-grained Transformation Module, complemented by a Hybrid Information Sharing mechanism to balance personalization and shared representation learning. FedMS[39] addresses privacy-preserving click-through rate prediction by combining a Slimify Module for model compression and a Feature Sharpening Module to enhance embedding

Table 1: Summary of Federated Learning Algorithm Optimization

Approach	**Global Model**	**Model Update**	**Local Training**	**Client Selection**
APFL [9]	×	×	✓	×
PCTPF [18]	✓	✓	✓	×
TiFL [16]	×	×	×	✓
FedDistill [17]	✓	×	✓	×
FedGH [22]	✓	×	×	×
LG-FedAvg [21]	×	✓	×	×
FedRep [24]	×	✓	✓	×
C-PFL	✓	✓	✓	✓

Table 2: Comparison of representative FL methods in terms of model architecture, training strategy, and aggregation mechanism

Method	**Model Architecture**	**Training Strategy**	**Aggregation Mechanism**
FedAvg	Monolithic global model	Train full model on local data	Weighted average of all client updates
FedGH	Global backbone + shared head	Full model training per round	Aggregates shared head with class-wise weights
FedProto	Shared encoder + class prototypes	Local training guided by prototypes	Server aggregates class prototypes
C-PFL	Shared backbone + local head	Backbone-only training during FL phase; local head fine-tuned post FL	Committee-based validation before aggregation; qualified backbone updates only

importance, achieving high accuracy with reduced computation and communication overhead.

These recent methods reflect the rapid progress in trustworthy and adaptive FL. Compared to them, C-PFL is distinguished by its emphasis on committee-driven robustness and architecture-level personalization.

3. Motivation

In federated learning (FL), multiple participants train a global model collaboratively without directly exchanging their local data. While this approach effectively preserves data privacy, it also introduces several challenges, such as ensuring the quality of local model updates and defending against potential malicious behavior or attacks. To address these issues, this work presents a *committee-based verification mechanism*, designed to filter and evaluate local updates submitted by clients. This mechanism helps eliminate low-quality or adversarial updates, thereby enhancing the performance, security, and reliability of the global model.

Due to variations in data distributions and computational resources among participants, the quality of the locally trained models can differ significantly. For instance, some clients may possess limited data, suffer from non-uniform data distributions, or fail to perform sufficient local training iterations, resulting in suboptimal model updates. If such updates are incorporated into the global model without proper screening, they may degrade overall performance.

A representative example can be found in medical applications, where data distributions differ across hospitals. A hospital with narrowly scoped data may produce a local model update that fails to generalize to other hospitals, ultimately hindering the performance of the global model. The committee mechanism assesses the effectiveness of these updates and ensures that only those that contribute positively to global performance are aggregated. This process mitigates the negative impacts of data imbalance and contributes to a more stable and generalizable global model.

One of the core challenges in FL is maintaining the global model's security while enabling collaborative training across decentralized participants. Given the open and distributed nature of FL, some clients may unintentionally or maliciously submit harmful model updates. Adversaries, for example, may launch poisoning attacks by deliberately modifying local models to corrupt the global model during aggregation.

3.1. Model separation

To defend against such threats, C-PFL adopts a partial update strategy, where clients submit only the *base* part (i.e., feature extractor) of their models during training. This limits the influence of malicious clients on other participants' models. By preserving the classification head locally and only aggregating the shared base, the robustness of the model is enhanced.

Nasr et al. [40] demonstrate that membership inference attacks can exploit gradients or parameters in the final layers to identify whether a specific data sample was used in training. By keeping the classification head entirely local, C-PFL effectively reduces exposure to these types of inference attacks.

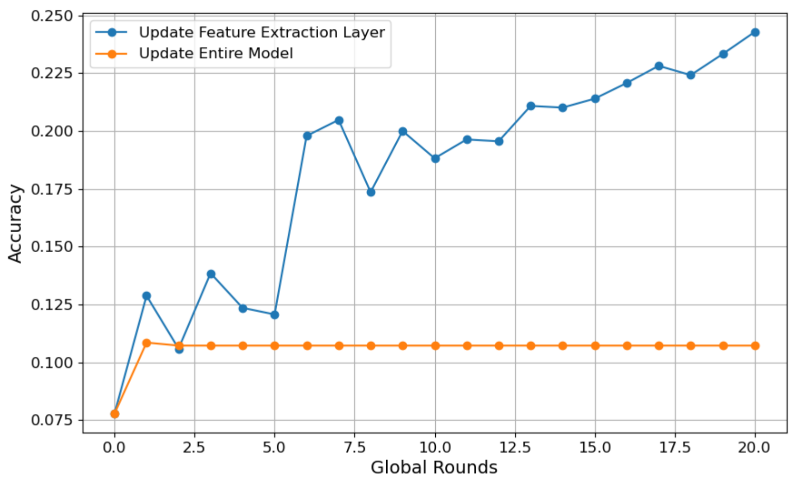


Figure 2: Accuracy comparison between feature extraction layer and entire model update.

Since the head never leaves the client device, an attacker observing shared model updates or gradients cannot directly reconstruct label distributions or sample identities, thus improving protection against both membership inference and label leakage attacks.

3.2. Committee Mechanism

The committee performs inspection and evaluation on each model update submitted by participating clients, aiming to identify updates that exhibit potentially adversarial or abnormal behavior. The committee rejects suspicious or malicious updates, preventing them from being incorporated into the global model aggregation process and thereby safeguarding the model from adversarial interference.

Beyond defending against intentional attacks, the committee mechanism is also effective in mitigating unintentional errors or faults. For example, updates from clients with low-quality data or hardware malfunctions may deviate significantly from the expected learning trajectory. In such cases, the committee serves as a filtering and correction mechanism, ensuring that only reliable updates contribute to the global model, thus maintaining its integrity and stability.

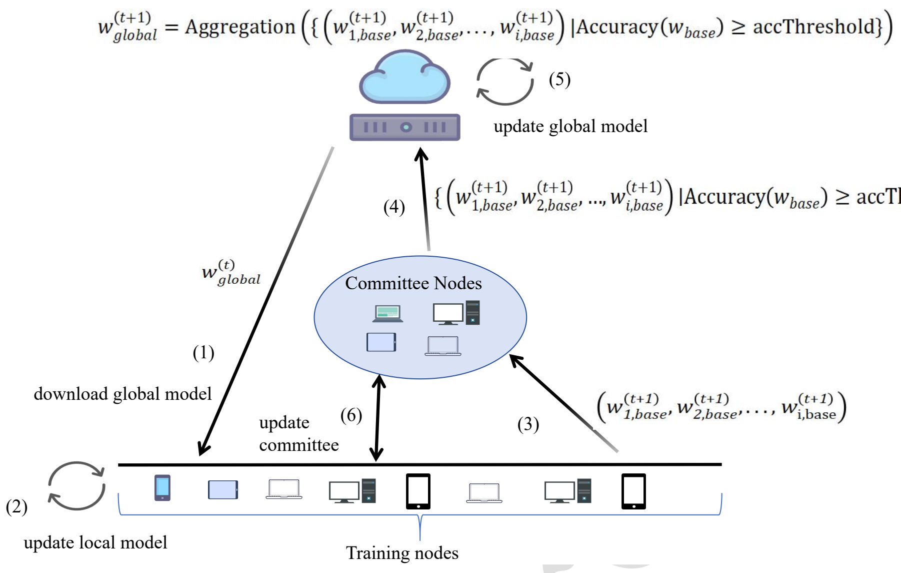

$w_{global}^{(t+1)}$ = Aggregation $\left(\left\{\left(w_{1,base}^{(t+1)}, w_{2,base}^{(t+1)}, \ldots, w_{i,base}^{(t+1)}\right) \mid \text{Accuracy}(w_{base}) \geq \text{accThreshold}\right\}\right)$
(5)
update global model
(4)
$\left\{\left(w_{1,base}^{(t+1)}, w_{2,base}^{(t+1)}, \ldots, w_{i,base}^{(t+1)}\right) \mid \text{Accuracy}(w_{base}) \geq \text{accThreshold}\right\}$
$w_{global}^{(t)}$
Committee Nodes
(1)
download global model
update committee
(6)
(3)
$\left(w_{1,base}^{(t+1)}, w_{2,base}^{(t+1)}, \ldots, w_{i,base}^{(t+1)}\right)$
(2)
update local model
Training nodes

Figure 3: Framework of C-PFL.

4. C-PFL algorithm

This section introduces the C-PFL algorithm, designed for federated learning with non-IID data. The C-PFL workflow is illustrated in Figure 3:

1. **Global Model Broadcast**: At the beginning of each training round, the server broadcasts the current global model $w_{\text{global}}^{(t)}$ to all training nodes.

2. **Local Model Training**: Each training node (i.e., device or client) uses its local dataset to train the received global model, producing an updated local model $w_{\text{i}}^{(t+1)}$.

3. **Submission of Local Model**: Each training node submits the base part of its updated local model $w_{\text{i,base}}^{(t+1)}$ to the committee nodes.

4. **Filtering and Evaluation**: The committee nodes evaluate the local models received based on a predefined performance threshold:

   $Accuracy(w_{base}) \geq accThreshold$. This evaluation is conducted using a validation set on the committee nodes. Only models that meet or exceed the threshold are considered for updating the global model. Those that fail the threshold are discarded and excluded from aggregation.

5. **Global Model Update**: On the server side, selected local models are aggregated to produce a new global model $w_{\text{global}}^{(t+1)}$. This new model will be sent to all training nodes in the next iteration to start a new training cycle.

6. **Committee Update**: After each round of federated learning, the committee nodes are also updated to prepare for the next round of model evaluations.

4.1. Model verification

In C-PFL, the committee must verify local models submitted by clients in each round before they can be used in global model aggregation. Suppose there are N clients, each owning a private training dataset $D_1, D_2, \ldots, D_N$, which are heterogeneous (non-IID and imbalanced). Among these N clients, M serve as training nodes and C as committee nodes, with $N = M + C$.

For a local model $w_i^{(k)}$ submitted by training node i in round k, the committee nodes evaluate it using their local data and produce validation accuracies $a_1, a_2, \ldots, a_C$. The local model is only eligible for global aggregation if the final score A_i meets the threshold $A_i \geq accThreshold$. The final score A_i is computed as:

$$A_i = \text{Average}(a_1, a_2, \ldots, a_C) \tag{1}$$

Using the average ensures a more robust assessment of the overall performance of the local model, reducing the risk of biased evaluations caused by the extreme results of the individual committee nodes. In a heterogeneous data environment, committee nodes usually have diverse data distributions and class compositions, significantly affecting model validation.

For instance, suppose committee node C_j primarily handles classification tasks for class t. If training node i's model $w_i^{(k)}$ is trained on data lacking samples of class t, the model may perform poorly when validated on C_j's data, resulting in a low accuracy score. This scenario often arises due to imbalanced or non-diverse training data.

In contrast, if the data distribution of the training node i closely matches that of the committee node C_j, particularly with a large proportion of class t, the model may achieve higher precision during validation. However, this could reflect class-specific advantages rather than true general performance across the global dataset.

By averaging validation results from multiple committee nodes, the evaluation process becomes more balanced and comprehensive, effectively mitigating the influence of individual data biases and ensuring that models are better suited to diverse data distributions.

4.2. Model update

In C-PFL, each client model is explicitly divided into two components: a shared feature extractor (the backbone) and a personalized classifier (the head). Formally, the full model $f(x)$ can be expressed as:

$$f(x; w_{\text{base}}, w_{\text{head}}) = h(x; w_{\text{head}}) \circ g(x; w_{\text{base}}),$$

where:

- w_{base} denotes the parameters of the backbone $g(\cdot)$, which consists of the convolutional and intermediate fully connected layers responsible for general feature extraction;
- w_{head} denotes the parameters of the head $h(\cdot)$, which performs classification and is tailored to each client's local data distribution;
- $\circ$ denotes function composition, i.e., the output of g is used as the input to h, so $f(x) = h(g(x))$.

Prior studies have shown that training the entire model or fixing only the head yields comparable performance, whereas fixing the base significantly degrades performance [6]. Therefore, in the local training phase, the head remains unchanged, and training nodes only update the base part of the model. After each training round, only the weights of the base are submitted. This approach enables the efficient training of a high-quality global base model.

During the federated training phase, only the shared backbone parameters w_{base} are updated collaboratively across clients. The head w_{head} remains fixed and is updated only during local fine-tuning after the global model convergence. This design enables effective personalization while maintaining a stable and generalizable global representation.

Once global training is complete, the head of the local model is no longer fixed. Each node can fine-tune the head using local data, achieving personalized adaptation. This strategy leverages the globally shared base for general feature extraction, allowing the head to be optimized locally. As a result, the model can better fit the specific data distribution of each client, improving both personalization and overall performance.

The C-PFL update procedure is outlined in Algorithm 1. The server first initializes the global model $w^{(0)}_{\text{global}}$ and the initial committee $C^{(0)}$ (Lines 1-2). Then, the selected training nodes perform local updates based on the global model (Lines 4-6). Each node trains locally and submits its updated base part. Afterward, the committee verifies the submitted local models (Lines 7-9), filtering out those that do not meet the criteria.

The verification process by the committee includes:

1. **Model Loading**: Committee node C_j loads the submitted model $w^{(i)}_{base}$.

2. **Local Inference**: The model $w^{(i)}_{base}$ is evaluated on the local dataset D_j, and the performance metric is calculated as:

$$V_j(w^{(i)}_{base}, D_j) = \text{Acc}(w^{(i)}_{base}, D_j) \tag{2}$$

$$\text{Acc}(w^{(i)}_{base}, D_j) = \frac{1}{N_j}\sum_{k=1}^{N_j} \mathbb{I}\left(f_{w^{(i)}_{base}}(x_k) = y_k\right) \tag{3}$$

   where Acc denotes the accuracy function. $f_{w^{(i)}_{base}}(x_k)$ denotes the prediction (i.e., the `argmax` output) of model $w^{(i)}_{base}$ on the input x_k. The function $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the condition inside holds true, and 0 otherwise. N_j represents the number of data samples on the local dataset D_j of committee node C_j. This equation reflects the proportion of correctly predicted samples on the local data of the committee node.

3. **Result Aggregation**: Each committee node C_j sends its evaluation result V_j to an aggregator, which computes the average score.

4. **Decision**: If the average score exceeds a predefined threshold τ, the model $w^{(i)}_{base}$ passes the verification and is included in the global model update; otherwise, it is rejected.

$$w^{(i)}_{qualified} \leftarrow \begin{cases} w^{(i)}_{base}, & \text{if } \frac{1}{m}\sum_{j=1}^{m} V_j(w^{(i)}_{base}, D_j) \geq \tau \\ \varnothing, & \text{otherwise} \end{cases} \tag{4}$$

where $w_{qualified}^{(i)}$ denotes the qualified backbone update from client i that will be used in aggregation, and $w_{base}^{(i)}$ is the backbone update submitted by client i. The function $V_j(w_{base}^{(i)}, D_j)$ measures the validation accuracy of $w_{base}^{(i)}$ on committee member j's local validation dataset D_j, and m is the total number of committee members in the current round. The parameter τ represents the current accuracy threshold, dynamically updated as described in Eq. 4. If the mean validation accuracy across all committee members meets or exceeds τ, the update is accepted; otherwise, it is discarded ($\varnothing$).

The server updates the global model by performing a weighted aggregation of the qualified local models based on the amount of data each client possesses (Line 10). Since the C-PFL aggregation process focuses solely on the feature extraction component of the model, the weight assigned to each local model is proportional to the size of the client's local dataset. This ensures that clients with more data have a greater influence on the global model and contribute more significantly to its update. The aggregation formula is given by:

$$W_{global}^{(i+1)} \leftarrow \sum_{k \in Client(q)} \frac{n_k}{n} \omega_{k,qualified}^{(i)} \tag{5}$$

where:

- $W_{global}^{(i+1)}$ represents the parameters of the global model after round i.
- $\omega_{k,qualified}^{(i)}$ denotes the local model parameters from client k in round i that passed the accuracy threshold.
- n_k is the size of the local dataset on client k.
- n is the total number of data samples across all clients, i.e., $n = \sum_{k=1}^{K} n_k$.
- $Client(q)$ denotes the set of clients whose models met the qualification criteria.

During local training, each node only updates and submits the base part of the model, which improves training efficiency and reduces communication overhead, as described in Algorithm 2.

Algorithm 1 C-PFL:Server

```
Input: global_rounds
Output: $w_{\text{global}}^{(\texttt{global\_rounds})}$
1.Initialize global model with parameters $w_{\text{global}}^{(0)}$
2.Initialize committee $C^{(0)}$
3.for $i = 0$ to global_rounds do
    4. $Client^{(i)} \leftarrow selectClients()$
    5. for each client $k \in Client^{(i)}$ in parallel do
        6. $w_{k,base}^{(i)} \leftarrow clientBaseUpdate_i(w_{\text{global}}^{(i)})$
        7. if $\frac{1}{m}\sum_{j=1}^{m} V_j(w_{k,base}^{(i)}, D_j) \geq \tau$ then
            8. $w_{k,qualified}^{(i)} \leftarrow w_{k,base}^{(i)}$
        end
        else
            9. $w_{k,qualified}^{(i)} \leftarrow \varnothing$
        end
    end
    10. $W_{\text{global}}^{(i+1)} \leftarrow \sum_{k \in Client(q)} \frac{n_k}{n} \omega_{k,qualified}^{(i)}$
end
11. Return $w_{\text{global}}^{(\texttt{global\_rounds})}$
```

4.3. Committee and threshold update

After the global model has been updated, it is necessary to update both the committee and the accuracy threshold. At the end of each round of training, a new committee S is selected from the set of clients whose local models have been verified. When each training node submits its local model, a score is calculated based on its data volume and accuracy. The score is recorded and accumulated round by round to form a cumulative contribution score that reflects the overall contribution of the node. New committee members will be selected from clients with the highest cumulative scores.

This strategy ensures that the committee is composed of nodes that have performed well in previous rounds, thereby improving the reliability and quality of model evaluation. In addition, this dynamic selection mechanism incentivizes clients to continuously improve their models to increase their

Algorithm 2 C-PFL:Client

```
Input: $w_{\text{global}}$
Output: $w_{feat}$
1.Copy $w_{\text{global}}$ to $w_{feat}$
2.for $epoch = 1$ to $E$ do
    3. Shuffle the local dataset $D_i$
    4. Split $D_i$ into mini-batches $\{B_1, B_2, \ldots, B_m\}$
    5. for each mini-batch $B_j$ do
        6. Compute gradients: $g_i \leftarrow \nabla L(w_{local}, B_j)$
        7. Update only feature extractor parameters:
        $w_{\text{feat}} \leftarrow w_{\text{feat}} - \eta \cdot g_i|_{w_{\text{feat}}}$
    end
end
8. Return $w_{feat}$
```

likelihood of being selected for the committee.

$$\text{Clients} = \{C_1, C_2, \ldots, C_m\} \tag{6}$$

$$\text{Contribution}_i = \alpha \cdot \widetilde{\text{Acc}}_i + (1-\alpha) \cdot \widetilde{N}_i \tag{7}$$

$$\{C_{(1)}, C_{(2)}, \ldots, C_{(m)}\} = \text{sort}(\text{Clients}, \text{by} = \textit{Contribution}, \text{descend}) \tag{8}$$

$$S = \{C_{(1)}, C_{(2)}, \ldots, C_{(n)}\} \tag{9}$$

In the above formula, $\widetilde{\text{Acc}}_i$ denotes the normalized validation accuracy of client i, which ensures that accuracy values are mapped to a uniform scale between 0 and 1:

$$\widetilde{\text{Acc}}_i = \frac{\text{Acc}_i - \min_j \text{Acc}_j}{\max_j \text{Acc}_j - \min_j \text{Acc}_j} \tag{10}$$

Here, Acc_i represents the raw validation accuracy of client i, and $\min_j \text{Acc}_j$, $\max_j \text{Acc}_j$ denote the minimum and maximum accuracy scores among all clients in the current round, respectively.

Similarly, $\widetilde{N}_i$ represents the normalized local data size of client i, computed as:

$$\widetilde{N}_i = \frac{N_i}{\sum_j N_j} \tag{11}$$

where N_i is the number of local samples held by client i, and the denominator is the total data volume across all participating clients.

The hyperparameter $\alpha \in [0, 1]$ controls the relative importance of accuracy versus data volume in the final contribution score. A larger α favors model performance as measured by validation accuracy, while a smaller α gives more weight to the quantity of data provided. In our implementation, this formulation ensures that clients with valuable data distributions are not overlooked due to lower immediate accuracy.

The procedure for updating the model accuracy threshold is presented in Algorithm 3. First (Lines 1-3), the algorithm calculates the number of accuracy values k, their mean $\bar{x}$, and standard deviation s. Then (Lines 4-6), it computes the standard error SE and determines the lower confidence bound LB using the t-distribution critical value t_α at significance level α. Next (Line 7), the accuracy threshold is updated to the maximum of its current value and the calculated lower bound, ensuring statistically grounded adjustments. Finally (Line 8), the committee for the next round $S^{(i+1)}$ is selected as the top-n clients based on their contribution values.

Algorithm 3 Accuracy Threshold Update via Confidence Interval

```
Input: accuracies: List of accuracy values,
α: Significance level (e.g., 0.05),
Clients: List of clients with contribution_values,
n: Committee size
Output: Updated accuracy threshold, Selected committee $S^{(i+1)}$
 1.$k \leftarrow \texttt{len}(accuracies)$
 2.$\bar{x} \leftarrow \texttt{mean}(accuracies)$
 3.$s \leftarrow \texttt{std}(accuracies)$
 4.$SE \leftarrow s/\sqrt{k}$ ;                    // Standard error of mean
 5.$t_\alpha \leftarrow \texttt{t-dist}(k-1, \alpha)$ ;         // t-value for $\alpha$ significance
 6.$LB \leftarrow \bar{x} - t_\alpha \cdot SE$ ;                 // Lower confidence bound
 7.$accThreshold \leftarrow \max(accThreshold, LB)$
 8.$S^{(i+1)} \leftarrow \mathrm{Top}_n\,(\mathrm{sort}(Clients, \mathrm{by} = \mathrm{contribution\_value}))$
```

The threshold update leverages the confidence interval theory: $P(\mu \geq \bar{x} - t_\alpha \frac{s}{\sqrt{k}}) = 1 - \alpha$ where μ is the true mean accuracy. This provides probabilistic guarantees that the updated threshold reflects genuine performance improvements. The t-distribution accounts for sample size effects, with $k-1$ degrees of freedom ensuring robustness for small committees.

In C-PFL, the committee reaches consensus through a quantitative evaluation process. Specifically, each committee node validates the submitted backbone update on its local dataset, producing an accuracy score. These scores are averaged across all committee nodes (Eq. 1) and compared with the dynamically updated threshold (Eq. 4). An update is accepted only if the averaged score meets or exceeds this threshold; otherwise, it is rejected. Clients whose updates are unstable or noisy due to training or hardware limitations may be excluded from aggregation. C-PFL can optionally provide clients with feedback on their evaluation results, including per-committee-node accuracy scores and whether the final decision was above or below the threshold.

To avoid evaluation bias and ensure that verification results generalize to the entire population, the data distribution across the selected committee nodes should collectively cover the diversity of features and labels present in the federation. In our implementation, this is promoted by the dynamic committee selection strategy, which ranks clients by both validation accuracy and data volume, indirectly favoring clients with more diverse and representative local datasets.

Moreover, C-PFL mitigates residual bias through the final local personalization phase: after global training, each client fine-tunes its personalized head on its local data. This step allows the model to adapt to unique local distributions that may not be fully represented in the committee's validation process, thereby reducing bias and improving overall generalizability in highly heterogeneous settings.

5. Experiments

5.1. Experimental setup

In this section, we compare C-PFL with six state-of-the-art federated learning methods—FedAvg [3], FedGH [22], Ditto [31], GPFL [23], FedProto [19], and LG-FedAvg [21]—on four benchmark datasets.

5.1.1. Datasets

To evaluate the performance of C-PFL on various classification tasks, we conducted experiments on four widely-used real-world image datasets: MNIST [41], Fashion-MNIST [42], CIFAR-10 [43], and CIFAR-100 [44].

- **MNIST**: This dataset consists of 70,000 grayscale images of handwritten digits (0–9), with 60,000 images for training and 10,000 for testing. Each image is of size 28×28 pixels.
- **Fashion-MNIST**: This dataset contains 70,000 grayscale images across 10 fashion categories (e.g., shirts, trousers, shoes), including 60,000 training and 10,000 testing samples. Each image is 28×28 pixels.
- **CIFAR-10**: The CIFAR-10 dataset contains 60,000 color images across 10 classes (e.g., airplane, car, bird), with 50,000 for training and 10,000 for testing. Each image is 32×32 pixels.
- **CIFAR-100**: This dataset comprises 60,000 color images divided into 100 categories. Each class contains 600 images, with 50,000 used for training and 10,000 for testing. Image size is 32×32 pixels.

To simulate real-world scenarios, we applied two types of non-IID (non-independent and identically distributed) data partitioning: practical non-IID and pathological non-IID. In practical non-IID, the distribution of labels across clients remains diverse but skewed — each client has access to most labels, but with biases toward certain classes. For example, in MNIST, each client may have a bias toward specific digits, but samples from other classes are still included. In pathological non-IID, clients receive data from only a single label class: for example, client 1 has only digit '0', client 2 only '1', etc. This extreme scenario reflects the worst-case heterogeneity in federated learning.

5.1.2. Software and Hardware Environment

Experiments were conducted on a system running Windows 10 with Python 3.9 and PyTorch 1.13.1 (with CUDA 11.7 support). The hardware setup includes an Intel i5-13400KF CPU (4.60 GHz), an NVIDIA RTX 4060 GPU (8 GB), and 32 GB of RAM.

5.1.3. Model Architecture

We employed a Convolutional Neural Network (CNN) as the base classifier for federated training. The model comprises two convolutional layers with 5×5 kernels — the first with 32 channels and the second with 64 channels. Each convolutional layer is followed by a 2×2 max-pooling layer with a stride of 1 to reduce feature map dimensions. A fully connected layer with 512 units is used for feature extraction, followed by a softmax layer for classification. The fully convolutional and connected layers form the base part of the model, while the softmax layer represents the head. To simplify comparison, the same architecture is used across MNIST, Fashion-MNIST, and CIFAR-10 tasks (with necessary adjustments to input dimensions).

The CNN model used in C-PFL is structured as follows:

$$f(x; w_{\text{base}}, w_{\text{head}}) = h(x; w_{\text{head}}) \circ g(x; w_{\text{base}}), \tag{12}$$

where:

- $g(x; w_{\text{base}})$ is the shared backbone network, consisting of:
  - `conv1`: Conv2D(1→32, kernel=5) + ReLU + MaxPool
  - `conv2`: Conv2D(32→64, kernel=5) + ReLU + MaxPool
  - `fc1`: Linear(1024→512) + ReLU
- $h(x; w_{\text{head}})$ is the personalized classification head:
  - `fc`: Linear(512→10)

During the federated training phase, only the parameters w_{base} of the backbone are exchanged and updated across clients. The head parameters w_{head} are frozen during global training and fine-tuned locally in a post-training phase to enable personalization.

5.1.4. Training Configurations

The number of participating clients in federated learning experiments ranges from 50 to 100. To emulate realistic and challenging conditions, datasets were partitioned into non-IID distributions, where the number of samples and label distribution vary randomly across clients. The learning rate is set to 0.005, and each client performs 5 local updates per communication round with a batch size of 10.

5.2. Normal training experiment

C-PFL consistently outperforms existing baseline methods under practical and pathological non-IID settings, across experiments with 50 and 100 clients.

Table 3: Average test accuracy (%) in homogeneous FL setting with 50 clients

Approach	Practical non-IID				Pathological non-IID			
	MNIST	FMNIST	CIFAR10	CIFAR100	MNIST	FMNIST	CIFAR10	CIFAR100
FedAvg	98.37	87.50	61.90	28.23	97.55	82.56	61.50	28.68
FedGH	98.94	97.36	88.45	46.40	99.61	99.33	87.60	42.18
Ditto	98.91	97.27	88.03	44.18	99.50	99.26	87.46	40.51
GPFL	95.83	95.61	84.59	48.09	99.77	98.90	86.29	41.85
FedProto	99.12	97.62	89.08	49.15	99.68	99.38	88.74	44.37
LG-FedAvg	98.94	97.29	88.13	44.45	99.57	99.29	87.76	40.54
FedALA	98.49	97.51	89.53	50.56	99.74	99.53	89.16	44.49
FedGC	98.91	97.20	88.14	42.99	99.70	99.48	88.76	43.64
C-PFL	**99.50**	**97.81**	**90.56**	**52.04**	**99.79**	**99.41**	**90.15**	**46.92**

Table 4: Average test accuracy (%) in homogeneous FL with 100 clients

Approach	Practical non-IID				Pathological non-IID			
	MNIST	FMNIST	CIFAR10	CIFAR100	MNIST	FMNIST	CIFAR10	CIFAR100
FedAvg	98.94	89.28	60.26	26.45	97.43	81.92	59.53	28.57
FedGH	98.32	95.64	86.69	43.96	99.51	99.19	85.21	39.65
Ditto	98.14	95.72	86.20	40.88	99.37	98.96	85.22	37.31
GPFL	99.17	94.93	85.73	47.17	99.77	99.00	85.55	38.36
FedProto	98.55	98.54	86.72	46.02	99.58	99.22	86.41	41.58
LG-FedAvg	98.19	95.69	86.24	41.12	99.47	99.20	85.27	37.68
FedALA	99.26	96.16	87.76	47.83	99.80	99.13	87.41	42.49
FedGC	98.54	96.53	87.44	39.56	99.76	99.23	87.16	41.80
C-PFL	**99.32**	**96.48**	**89.42**	**50.49**	**99.82**	**99.21**	**89.49**	**44.48**

In the practical non-IID setting, C-PFL achieves a precision of 90.56% and 52.04% in CIFAR-10 and CIFAR-100, respectively. This represents improvements of 1.48% and 2.89% over FedProto (89.08% and 49.15%), and 2.11% and 5.64% over FedGH (88.45% and 46.40%). In the pathological non-IID scenario, C-PFL attains 90.15% precision in CIFAR-10 and 46.92% in CIFAR-100. These results outperform FedProto (88.74% and 44.37%) by 1.41% and 2.55%, and GPFL (86.29% and 41.85%) by 3.86% and 5.07%, respectively.

When the number of clients increases to 100, C-PFL still maintains superior performance. In the practical non-IID CIFAR-10 and CIFAR-100 datasets, C-PFL achieves 89.42% and 50.49% precision, surpassing FedProto (86.72% and 46.02%) by 2.70% and 4.47%. Compared with FedGH (86.69% and 43.96%), the improvements are 2.73% and 6.53%, respectively, highlighting the stronger generalization capability of C-PFL. In the pathological setting without IID, C-PFL records the precision of 89.49% and 44.48% in CIFAR-10 and CIFAR-100, respectively, which significantly outperform Fed-

Proto (86.41% and 41.58%) by 3.08% and 2.90%, and Ditto (85.22% and 37.31%) by 4.27% and 7.17%.

In particular, C-PFL shows strong robustness, with minimal performance fluctuation across varying degrees of data heterogeneity. Even under highly pathological non-IID conditions, C-PFL consistently maintains superior accuracy. In complex datasets such as CIFAR-100, C-PFL achieves accuracy around 50%, while traditional methods such as FedAvg and Ditto often fall below 30% or 40%.

These results collectively indicate that C-PFL delivers higher classification performance and ensures greater stability in diverse federated learning scenarios.

While C-PFL has been evaluated with up to 100 clients under synchronous training, its scalability to significantly larger federations (e.g., thousands of clients) and applicability in fully asynchronous environments have not yet been tested. A promising direction to address these challenges is to adopt *committee sampling* or *partial consensus* strategies, where only a dynamically sampled subset of committee members participates in each verification round, or consensus is reached based on a majority subset rather than all committee members. We plan to explore these extensions in future work to further enhance the scalability and flexibility of C-PFL.

5.3. Robustness to malicious clients

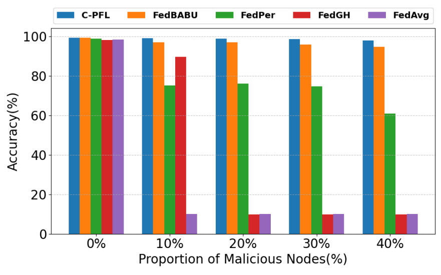
To evaluate the robustness of C-PFL against malicious or faulty clients, we simulate *noise-injection attacks*, where a subset of clients adds Gaussian noise to their input data during local training. This method preserves the original labels but corrupts the input distribution, mimicking real-world scenarios such as sensor failures, adversarial corruption, or low-quality data acquisition.

Specifically, in each communication round, we randomly select a fixed percentage of clients (10%, 20%, 30%, or 40%) to act as noisy participants. For these clients, we apply additive Gaussian noise to each local sample:

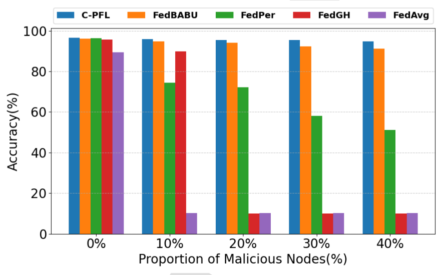
$$x_{\text{noisy}} = x + \mathcal{N}(\mu, \sigma^2) \tag{13}$$

where μ and σ denote the noise mean and standard deviation, respectively. In our experiments, we set $\mu = 0$ and vary $\sigma \in \{0.1, 0.2, 0.3\}$ to simulate increasing levels of corruption.

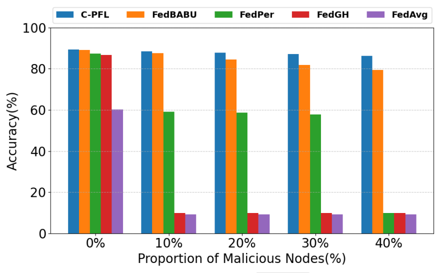
This attack model has been used in prior federated learning studies (e.g., [45, 46]) to evaluate model robustness under non-malicious but harmful local



(a) MNIST



(b) Fashion-MNIST



(c) CIFAR-10

Figure 4: Performance of methods under the influence of noise on different datasets.

updates. By introducing this noise-based corruption, we assess whether C-PFL can effectively identify and mitigate the influence of degraded or low-quality model contributions.

Figure 4 presents the performance degradation of various federated learning methods on three datasets (MNIST, Fashion-MNIST, and CIFAR-10) as the proportion of malicious clients increases from 0% to 40%. Each sub-figure corresponds to a dataset, with the x-axis representing the percentage of malicious clients and the y-axis indicating the model accuracy. Across all data sets, C-PFL shows the smallest accuracy drop as the proportion of malicious clients increases, highlighting its strong resilience to adversarial noise. Even when 40% of the clients are malicious, C-PFL outperforms the best baseline by 3.12% on MNIST, 3.58% on Fashion-MNIST, and 6.96% on CIFAR-10.

In particular, on the CIFAR-10 dataset, C-PFL maintains stable performance with only minor degradation as the percentage of malicious nodes increases: accuracy decreases by just 1.04%, 1.68%, 2.31%, and 3.18% for 10%, 20%, 30%, and 40%, respectively. In contrast, the best-performing baseline, FedBABU, experiences more severe drops under the same conditions:

1.54%, 4.62%, 7.36%, and 9.89%, respectively. This underscores C-PFL's robustness, especially in scenarios with high adversarial presence. Methods such as FedGH and FedAvg show significantly worse robustness, with accuracy sharply declining beyond the proportion of malicious clients 10% and becoming nearly ineffective.

In this work, we restrict our robustness evaluation to label-flipping attacks. We do not include adaptive or backdoor attacks, primarily because such attacks often require task-specific trigger patterns to be realistically implemented. The backdoor trigger may lead to unfair or incomparable settings across methods. Nevertheless, we acknowledge that stealthy adversaries such as adaptive or backdoor attacks pose greater challenges, since they can evade detection by behaving normally on clean inputs. Extending C-PFL to defend against these stronger adversaries will be an important direction for future work.

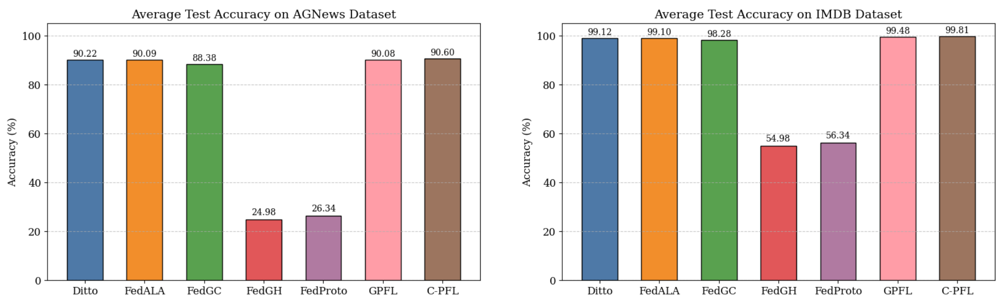
5.4. Evaluation on text data

To assess the generalizability of C-PFL beyond image datasets, we conducted experiments on the AGNews[47] dataset, a four-class news classification benchmark. Following a non-IID label distribution among clients, we compared C-PFL with six personalized FL baselines. As shown in Fig. 5, C-PFL achieves the best performance (90.60%). On IMDB[48], C-PFL achieves the highest accuracy of 99.81%, outperforming all six baselines. Compared to GPFL, which obtains the second-best performance of 99.48%, C-PFL yields an improvement of 0.33%.

We also note that FedGH and FedProto perform significantly worse on AGNews and IMDB. This can be attributed to their underlying assumptions: FedGH requires meaningful graph structures, which are not naturally present in textual data; FedProto relies on class-wise prototype representations in embedding space, which may be unstable in NLP tasks. In contrast, C-PFL does not impose such constraints and generalizes well across data modalities.

5.5. Overhead Analysis

While C-PFL introduces additional steps such as committee-based validation and contribution scoring, it is essential to examine whether these mechanisms incur prohibitive computational costs in typical FL settings. Table 5 presents the average time per global training round across three benchmark datasets.



(a) Average test accuracy (%) on AGNews dataset. (b) Average test accuracy (%) on IMDB dataset.

Figure 5: Average test accuracy (%) on text dataset.

Table 5: The time consumed by each global round of each approach.

Approach	MNIST (s)	Fashion-MNIST (s)	CIFAR-10 (s)
FedAvg	29.71	28.94	33.62
FedProto	85.15	83.32	78.84
FedGH	33.48	33.35	37.93
Ditto	44.07	45.58	53.73
GPFL	71.48	78.61	64.40
LG-FedAvg	36.69	36.43	37.69
C-PFL	**37.94**	**39.35**	**41.84**

Compared to FedAvg, the lightest baseline, C-PFL incurs additional time of 8.23s on MNIST (+27.7%), 10.41s on Fashion-MNIST (+36.0%), and 8.22s on CIFAR-10 (+24.5%). However, C-PFL remains significantly more efficient than methods such as FedProto (78–85s/round), GPFL (64–78s/round), and Ditto (45–53s/round), all of which introduce much higher round-level latency. It is also notable that C-PFL's overhead remains comparable to LG-FedAvg and FedGH, which do not offer the same level of robustness or malicious client defense. Given C-PFL's superior performance under adversarial settings, we consider this overhead a practical and acceptable trade-off.

Furthermore, since only the backbone parameters are shared and validated, the communication payload is slightly reduced compared to full-model-sharing schemes. Committee nodes perform only lightweight forward inference on local data, and the cumulative scoring mechanism involves negligible computational logic (i.e., accuracy recording and averaging).

5.6. Local training analysis

To evaluate the personalization efficiency of C-PFL, we study the effect of fine-tuning epochs during the evaluation phase. Table 6 summarizes the model accuracy in four datasets with different local fine-tuning settings. Results show that C-PFL can achieve effective personalization with minimal local training.

Specifically, significant accuracy gains are observed after only 5 epochs of fine-tuning. For example, on CIFAR-10, the precision improves from 53.86% to 89.15%, while on CIFAR-100 it increases from 26.09% to 49.34%. Beyond 10 epochs, most datasets exhibit diminishing returns, with accuracy improvements of less than 0.5% between 10 and 20 epochs. Notably, Fashion-MNIST shows a slight performance drop of 0.13% at 20 epochs, suggesting potential overfitting. In general, C-PFL achieves near-optimal personalized performance with as few as five local training epochs, effectively balancing personalization accuracy and computational efficiency, while minimizing the risk of overfitting.

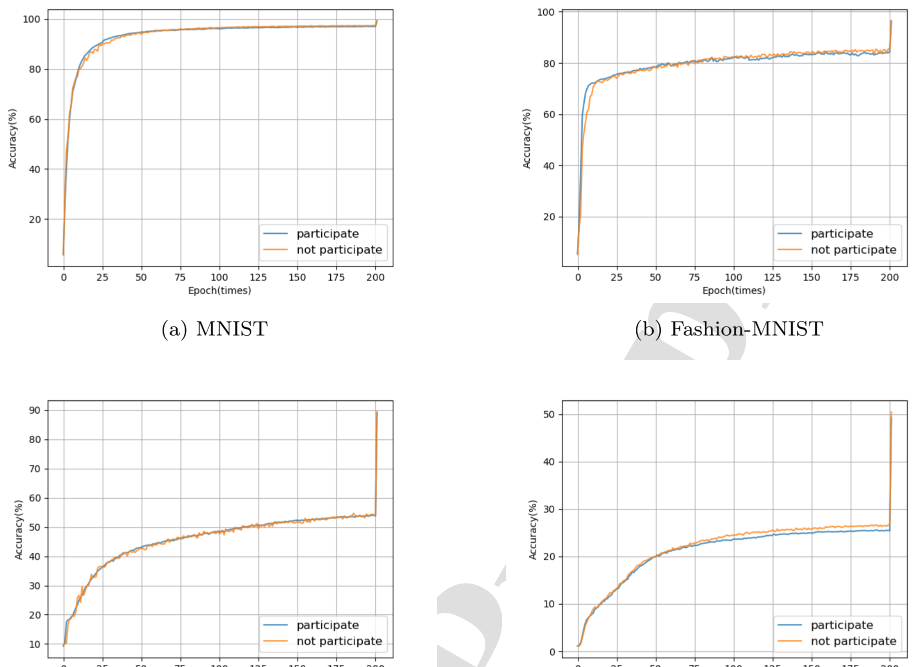
Table 6: Performance (%) with different fine-tuning rounds.

Dataset	**0 Epochs**	**5 Epochs**	**10 Epochs**	**15 Epochs**	**20 Epochs**
MNIST	97.46	99.27	99.28	99.29	99.32
Fashion-MNIST	85.42	96.16	96.33	96.44	96.31
CIFAR-10	53.86	89.15	89.30	89.34	89.30
CIFAR-100	26.09	49.34	50.19	50.25	50.21

5.7. Impact of committee selection on training

Figure 6 illustrates the impact of the participation of committee nodes in training on global model accuracy in four datasets. MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100. The figure compares two settings: the nodes of the committee that participate in training (the blue curve represents those that participate in the training process, and the orange curve represents those that do not participate in the training process). The accuracy curves under both settings are nearly identical, exhibiting consistent trends and converging to similar final performance.

These results indicate that, under our experimental configuration, whether committee nodes participate in local training has a negligible influence on the overall model performance.



(a) MNIST (b) Fashion-MNIST

(c) CIFAR-10 (d) CIFAR-100

Figure 6: Impact of committee node participation in training on model performance across different datasets.

This phenomenon can be attributed to two main reasons. First, committee nodes constitute a relatively small fraction of the total client population, and thus their contribution to model aggregation is limited. Second, the federated aggregation mechanism is inherently robust to individual node participation. It effectively integrates information from most regular clients to maintain stable global performance. As a result, even when committee nodes are restricted to evaluation and voting without local training, the overall model accuracy remains largely unaffected.

6. Conclusion

This paper presents C-PFL, a personalized federated learning approach featuring a committee-based validation mechanism that significantly bolsters model robustness and personalization in non-IID and adversarial environments. By leveraging dynamically selected committee nodes to validate and filter local model updates, C-PFL effectively discards low-quality or malicious contributions without relying on public datasets, thereby enhancing the stability and generalization of the global model. In addition, the model architecture of the method distinctly separates the base and personalized layers, allowing clients to preserve shared federated knowledge while catering to local personalization needs, thus adjusting easily to diverse data distributions between clients.

Extensive experiments on multiple real-world datasets demonstrate that C-PFL outperforms state-of-the-art methods in both accuracy and noise resilience. In particular, it maintains high performance even under high proportions of malicious client attacks, underscoring its practical applicability and reliability. Furthermore, the experimental results confirm that the participation of the committee nodes in training minimally affects the overall performance, offering a theoretical foundation to reduce their computational overhead.

6.1. Future work

In future work, we plan to extend C-PFL in two directions. We aim to explore dynamic committee scheduling mechanisms that adaptively adjust member selection based on evolving client behavior or data characteristics. Combining our framework with differential privacy techniques, such as DP-SGD or secure aggregation, could enhance client-side privacy while preserving robustness. These extensions will further enhance the applicability and trustworthiness of personalized federated learning in practical deployments. Furthermore, the present evaluation focuses exclusively on image and text datasets because our backbone–head split design may not fully capture the contextual dependencies inherent in natural language. Future work will assess the effectiveness of C-PFL on other data modalities, such as time-series or tabular datasets, to further validate its generalizability across diverse application domains.

Acknowledgments

This work is partially supported by the Natural Science Basic Research Program of Shaanxi (Program No.2025JC-YBMS-688), the Key R&D Programs of Taicang, 2024 under Grant TC2024SF10, and the XJTLU Research Development Fund under Grant No. RDF-22-02-106.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[2] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[3] H. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Arcas, "Communication-efficient learning of deep networks from decentralized data," *arXiv: Learning,arXiv: Learning*, Feb 2016.

[4] C. Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Neural Information Processing Systems,Neural Information Processing Systems*, Jan 2020.

[5] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization.," *International Conference on Artificial Intelligence and Statistics,International Conference on Artificial Intelligence and Statistics*, Sep 2019.

[6] J. Oh, S. Kim, and S.-Y. Yun, "Fedbabu: Towards enhanced representation for federated image classification," *arXiv preprint arXiv:2106.06042*, 2021.

[7] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 12, pp. 9587–9603, 2022.

[8] X.-C. Li, D.-C. Zhan, Y. Shao, B. Li, and S. Song, "Fedphp: Federated personalization with inherited private models," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 587–602, Springer, 2021.

[9] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," *arXiv preprint arXiv:2003.13461*, 2020.

[10] L. Li, M. Duan, D. Liu, Y. Zhang, A. Ren, X. Chen, Y. Tan, and C. Wang, "Fedsae: A novel self-adaptive federated learning framework in heterogeneous systems," in *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, IEEE, 2021.

[11] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, and H. Guan, "Fedala: Adaptive local aggregation for personalized federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11237–11244, 2023.

[12] Z. Ma, M. Zhao, X. Cai, and Z. Jia, "Fast-convergent federated learning with class-weighted aggregation," *Journal of Systems Architecture*, vol. 117, p. 102125, 2021.

[13] J. Luo and S. Wu, "Adapt to adaptation: Learning personalization for cross-silo federated learning," in *IJCAI: proceedings of the conference*, vol. 2022, p. 2166, NIH Public Access, 2022.

[14] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," *arXiv preprint arXiv:2012.08565*, 2020.

[15] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-iid data," in *Proceedings of the AAAI conference on artificial intelligence*, pp. 7865–7873, 2021.

[16] Z. Chai, A. Ali, S. Zawad, S. Truex, A. Anwar, N. Baracaldo, Y. Zhou, H. Ludwig, F. Yan, and Y. Cheng, "Tifl: A tier-based federated learning system," in *Proceedings of the 29th international symposium on high-performance parallel and distributed computing*, pp. 125–136, 2020.

[17] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," *arXiv preprint arXiv:1811.11479*, 2018.

[18] D. Wang, S. Guan, and R. Sun, "A novel staged training strategy leveraging knowledge distillation and model fusion for heterogeneous federated learning," *Journal of Network and Computer Applications*, p. 104104, 2025.

[19] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "Fedproto: Federated prototype learning across heterogeneous clients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8432–8440, 2022.

[20] J. Xu, X. Tong, and S.-L. Huang, "Personalized federated learning with feature alignment and classifier collaboration," *arXiv preprint arXiv:2306.11867*, 2023.

[21] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," *arXiv preprint arXiv:2001.01523*, 2020.

[22] L. Yi, G. Wang, X. Liu, Z. Shi, and H. Yu, "Fedgh: Heterogeneous federated learning with generalized global header," in *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 8686–8696, 2023.

[23] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, J. Cao, and H. Guan, "Gpfl: Simultaneously learning global and personalized feature information for personalized federated learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5041–5051, 2023.

[24] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International conference on machine learning*, pp. 2089–2099, PMLR, 2021.

[25] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[26] Y. Niu and W. Deng, "Federated learning for face recognition with gradient correction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1999–2007, 2022.

[27] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *International conference on machine learning*, pp. 12878–12889, PMLR, 2021.

[28] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Advances in neural information processing systems*, vol. 33, pp. 21394–21405, 2020.

[29] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10713–10722, 2021.

[30] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[31] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International conference on machine learning*, pp. 6357–6368, PMLR, 2021.

[32] L. Sun, S. Liu, Z. Qu, and Y. Zhang, "A federated learning-based zero-trust model with secure dynamic trust evaluation and knowledge transfer," *IEEE Journal on Selected Areas in Communications*, 2025.

[33] B. Lei, Y. Zhu, E. Liang, P. Yang, S. Chen, H. Hu, H. Xie, Z. Wei, F. Hao, X. Song, *et al.*, "Federated domain adaptation via transformer for multi-site alzheimer's disease diagnosis," *IEEE Transactions on Medical Imaging*, vol. 42, no. 12, pp. 3651–3664, 2023.

[34] A. Radwan and M. Shehata, "Fedpartwhole: federated domain generalization via consistent part-whole hierarchies," *Pattern Analysis and Applications*, vol. 28, no. 2, p. 61, 2025.

[35] S. Wang, Y. Fu, X. Li, Y. Lan, M. Gao, *et al.*, "Dfrd: Data-free robustness distillation for heterogeneous federated learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 17854–17866, 2023.

[36] Y. Di, H. Shi, X. Wang, R. Ma, and Y. Liu, "Federated recommender system based on diffusion augmentation and guided denoising," *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–36, 2025.

[37] Y. Di, H. Shi, R. Ma, H. Gao, Y. Liu, and W. Wang, "Fedrl: A reinforcement learning federated recommender system for efficient communication using reinforcement selector and hypernet generator," *ACM Transactions on Recommender Systems*, vol. 4, no. 1, pp. 1–31, 2025.

[38] Y. Di, X. Wang, H. Shi, C. Fan, R. Zhou, R. Ma, and Y. Liu, "Personalized consumer federated recommender system using fine-grained transformation and hybrid information sharing," *IEEE Transactions on Consumer Electronics*, 2025.

[39] Y. Di, H. Shi, J. Fan, J. Bao, G. Huang, and Y. Liu, "Efficient federated recommender system based on slimify module and feature sharpening module," *Knowledge and Information Systems*, pp. 1–34, 2025.

[40] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*, pp. 739–753, IEEE, 2019.

[41] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, p. 2278–2324, Jan 1998.

[42] X. Han, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv: Learning,arXiv: Learning*, Aug 2017.

[43] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Jan 2009.

[44] A. Krizhevsky, V. Nair, and G. Hinton, “Cifar-100 datasets.” `https://www.cs.toronto.edu/~kriz/cifar.html`, 2009. Accessed: 2025-04-19.

[45] M. Fang, X. Cao, J. Jia, and N. Gong, “Local model poisoning attacks to {Byzantine-Robust} federated learning,” in *29th USENIX security symposium (USENIX Security 20)*, pp. 1605–1622, 2020.

[46] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, “How to backdoor federated learning,” in *International conference on artificial intelligence and statistics*, pp. 2938–2948, PMLR, 2020.

[47] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” *Advances in neural information processing systems*, vol. 28, 2015.

[48] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Declaration of interests

☐ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Hao Guo reports financial support was provided by the Natural Science Basic Research Program of Shaanxi (Program No.2025JC-YBMS-688), the Key R&D Programs of Taicang, 2024 under Grant TC2024SF10. Wanxin Li reports financial support was provided by XJTLU Research Development Fund under Grant No. RDF-22-02-106. Hao Guo reports a relationship with Key R&D Programs of Taicang that includes: funding grants. Wanxin Li reports a relationship with XJTLU Research Development Fund under Grant No. RDF-22-02-106 that includes: funding grants. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.