Information Sciences 745 (2026) 123425

Contents lists available at ScienceDirect

Information Sciences

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

Dawid-Skene-model-based label-noise mitigation for federated learning

Jia Dong [a,*], Rui Zhu [b], Xinyi Shang [a], Jing-Hao Xue [a]

[a] Department of Statistical Science, University College London, London, WC1E 6BT, UK
[b] Bayes Business School, City St George's, University of London, London, EC1Y 8TZ, UK

ARTICLE INFO

Keywords:
Federated learning
Noisy labels
Dawid-Skene model
Client weighting

ABSTRACT

Federated learning (FL) enables collaborative model training without centralising raw data, but its performance is susceptible to label noise from clients. A common mitigation strategy involves using a clean, labelled public dataset at the server to assess client reliability. However, this approach is impractical due to the unrealistic assumption of availability of a clean, labelled public dataset. To address this issue, we propose FedDS, a novel approach that brings the Dawid-Skene model from statistical analysis to FL, which enables the estimation of the reliability of each client in FL without requiring any labelled data at the server. This approach effectively mitigates the adverse impact of heterogeneous label noise under a weaker and more practical assumption, offering a robust aggregation strategy for real-world FL scenarios with label noise. The code is available at https://github.com/Gia99999/FedDS.

1. Introduction

Federated learning (FL) has emerged as a promising paradigm for training machine learning models in a decentralised manner while preserving data privacy [1,2]. In this distributional paradigm of machine learning, individual clients maintain their local datasets and perform model training independently, sharing only the resulting individual model updates with a central server, and then this server aggregates the updates to form a global model [3–5]. However, one of the primary challenges in FL is the presence of label noise, which can arise from annotation errors or inherent ambiguities in the data [6,7]. Such noise can significantly deteriorate the performance of the final global model, particularly as label-noise levels are unknown and heterogeneous across clients in FL [8–11]. This has led to increasing research interest in federated learning with noisy labels (FLNL) [12].

In FLNL, a common approach to mitigating the influence of unreliable clients is to estimate client reliability at the server using a clean, labelled public dataset. Techniques such as client reweighting [13], pruning [14], and selection [15] rely on this dataset to compute performance indicators or statistical divergences of clients, thereby steering aggregation toward the clients deemed more trustworthy. However, the effectiveness of this approach depends on the often unrealistic assumption that the public dataset is entirely clean and accurately labelled.

Fig. 1 showcases the limitation of this unrealistic assumption. We systematically varied the label noise ratio in the server's public dataset and evaluated the resulting global model accuracy. Across these three methods, namely client reweighting [13], pruning [14], and selection [15], the accuracy exhibited a consistent downward trend as the label noise ratio increased, indicating that introducing label noise into the server's public dataset degrades global model accuracy. This decline arises because the corrupted labels distort

* Corresponding author.
Email address: jia.dong.23@ucl.ac.uk (J. Dong).

https://doi.org/10.1016/j.ins.2026.123425
Received 19 December 2025; Received in revised form 22 March 2026; Accepted 24 March 2026
Available online 25 March 2026

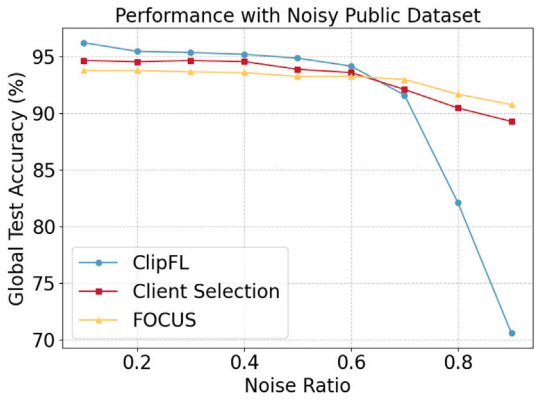

Fig. 1. The impact of a label-noisy public dataset at the server on the test accuracy of global model. The performance of all three methods, FOCUS [13], ClipFL [14] and Client Selection [15], degrades as the label noise ratio increases, highlighting the limitation of these methods due to their dependence on a clean dataset for the evaluation of client reliability.

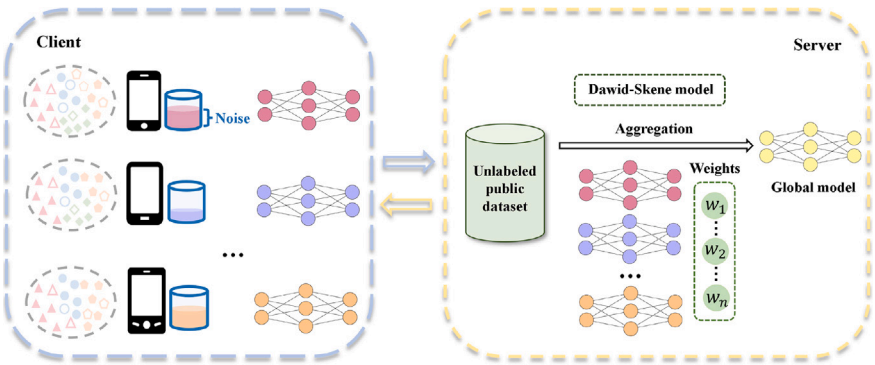

Fig. 2. Overview of the proposed FedDS approach. It borrows the strength of the Dawid-Skene model to estimate the reliability of each client without requiring any labelled public dataset at the server. The reliability estimates are then used to weight clients during global aggregation to mitigate the impact of noisy labels.

the estimates of client reliability, causing the server to misjudge client quality and make wrong aggregation decisions. In short, the results highlight the weakness of the methods that depend on server-side public datasets for the estimation of client reliability.

In addition to the vulnerability to label noise, several practical factors further limit the feasibility of maintaining a clean, labelled public dataset at the server. For instance, the construction of a clean, labelled dataset entails substantial annotation time and cost, particularly in fields requiring expert annotation. Furthermore, such a public labelled dataset is vulnerable to security threats and adversarial attacks, especially those inducing mislabelled samples, which can result in severe damage to the whole system of FL.

To overcome these limitations, we bring the Dawid-Skene model [16] to FLNL and propose a Dawid-Skene-model-based label-noise mitigation method (FedDS) to address heterogeneous label noise between clients. The diagram of FedDS is shown in Fig. 2. The Dawid-Skene model is a statistical model originally developed for aggregating various noisy annotations to recover the latent ground truth. It estimates both the true labels of data and the annotator-specific confusion matrices through an expectation-maximisation (EM) iterative process. In our setting, the various annotators are the heterogeneous label-noisy clients participating in FL. *By borrowing the strength of the Dawid-Skene model, we enable FLNL to operate without requiring any labelled public dataset at the server.*

Specifically, instead of requiring a clean, labelled public dataset, our method only needs an unlabelled public dataset hosted at the server. In each communication round, the Dawid-Skene model aggregates all individual client models' predictions of these samples to calculate client-specific confusion matrices. From each confusion matrix, we compute a reliability score for each client, reflecting its general predictive performance, even under heterogeneous label noise at the client level. Statistically, the estimation follows an EM procedure: in the expectation step, the posterior probabilities of the true labels for the public dataset are updated; in the maximisation step, these updated probabilities are used to refine the client-specific confusion matrices; and these two steps are iterated until convergence, producing a robust estimate of reliability for each client. These reliability scores are then used to weight clients during the global aggregation, so that less reliable clients have smaller influence on the aggregated model update in each communication round.

In summary, the core contributions of this paper are threefold from three perspectives. First, conceptually, we propose FedDS, a novel FLNL approach that leverages the Dawid-Skene model to estimate client-specific reliability without requiring any labelled public dataset at the server, by reformulating client model predictions on a server-side unlabelled public dataset as noisy annotations within a probabilistic framework. Secondly, technically, by integrating the reliability estimates from the Dawid-Skene model into the aggregation step of FLNL, our FedDS mitigates the adverse impact of noisy labels on the global model and provides a promising

solution to FLNL under unknown, heterogeneous label noise across local clients in a plug-and-play manner, without modifying the local training process. Thirdly, empirically, extensive experiments including ablation studies showcase the robustness and practical effectiveness of our method under weaker assumptions on the availability of a server-side public dataset.

2. Related work

2.1. FLNL methods

To address label noise in FL, existing FLNL methods can be broadly divided into three categories. Sample-wise approaches operate on individual clients to identify and handle noisy labels directly within their local datasets, for example, through sample selection [10,17,18] or label correction [19–21]. Client-wise approaches focus on determining which clients should participate in training [22] and how much each client should contribute to the global aggregation, often based on the estimated reliability or performance of each client [11,23,24]. Model-wise approaches aim to enhance robustness by modifying the training procedure so that the model can better tolerate noisy labels [25–28]. While each category of methods tackles the FLNL problem from a different perspective, all face their own limitations and trade-offs, particularly when applied in heterogeneous and privacy-constrained federated environments. Table 1 summarises some of these representative FLNL methods from sample-wise, client-wise, and model-wise perspectives, highlighting their research objectives and strategies.

2.2. Server-side public dataset in FL

Motivated by the need for assessing client behaviour without accessing private data in federated learning, a number of FLNL methods rely on a server-side public dataset as an auxiliary reference, some representative approaches of which are summarised in Table 2. Chen et al. [13] are among the first to propose using a clean, labelled public dataset to estimate client reliability without accessing client data. They combine the cross-entropy losses of each client's local model on the public dataset and its local data to compute client credibility. In Tuor et al. [29], a labelled public dataset is used to filter noisy data by comparing the loss distributions of client data with those of the public dataset. Clients whose data fall below a threshold are excluded from training, assuming the public dataset is clean. Wu et al. [30] use a public dataset to generate a baseline footprint for the global model and compute the Kullback-Leibler divergence between each client's local footprint and the baseline, adjusting client participation based on similarity, which similarly assumes that the public dataset is clean. Yang et al. [15] propose using a clean public dataset to infer the noise ratio of each client by evaluating the accuracy of client models on clean labels, ranking clients based on their accuracy and using this

Table 1
Summary of some representative FLNL methods, categorised by the level at which label noise is addressed, including sample-wise, client-wise, and model-wise approaches, along with their research objectives and strategies.

Category	Reference	Year	Research Objective	Strategy
Sample-wise	Han et al. [17]	2023	Identify clean samples from locally noisy data while avoiding the removal of hard but informative instances.	Use feature-difference scores from supervised contrastive learning.
	Ji et al. [10]	2024		Select low loss samples using alternating local and global models.
	Sun et al. [18]	2024		Incorporate samples from clean to noisy via curriculum learning.
	Xu et al. [19]	2022	Refine noisy labels to use noisy samples without propagating incorrect supervision.	Replace with pseudo labels inferred from loss-based GMMs.
	Zeng et al. [20]	2023		Correct mixed noise by selectively replacing closed-set noisy labels.
	Li et al. [21]	2024		Collaboratively aggregate noise models across clients.
Client-wise	Giap et al. [22]	2025	Exclude unreliable clients to prevent harmful updates.	Identify unreliable clients by treating new or unstable clients as noisy and refining their reliability over time.
	Zeng et al. [11]	2024	Estimate client reliability under heterogeneous label noise and adjust aggregation weights accordingly.	Assign weights based on parameter importance discrepancies.
	Xu et al. [23]	2024		Assign weights based on similarity to a clean reference model.
	Tsouvalas et al. [24]	2024		Assign weights based on energy scores derived from model outputs.
Model-wise	Yu et al. [25]	2025	Mitigate model overfitting to noisy labels caused by local memorisation under limited client data.	Apply forward loss correction using a noise transition matrix.
	Pu et al. [26]	2025		Delay memorisation of noisy labels via early-learning regularisation.
	Ejigu et al. [27]	2025		Use relaxed contrastive learning with local-global feature alignment.
	Ejigu et al. [28]	2025		Employ feature regularisation and sharpness-aware optimisation.

Table 2
Summary of some FL methods that leverage a server-side public dataset to mitigate label noise, highlighting their reliance on clean labels, research objectives and strategies.

Reference	Year	Label	Research Objective	Strategy
Chen et al. [13]	2020	Clean	Estimate client reliability.	Compute client credibility using losses on clean public data.
Tuor et al. [29]	2021	Clean	Filter noisy client data.	Filter samples by comparing loss distributions with clean public data.
Wu et al. [30]	2023	Clean	Identify unreliable clients.	Measure divergence between local and global outputs on clean public data.
Yang et al. [15]	2021	Clean	Select clients.	Rank clients by accuracy evaluated on clean public data.
Morafah et al. [14]	2025	Clean	Prune noisy clients.	Exclude clients based on accuracy on clean public data.
Lu et al. [31]	2024	No	Identify severely noisy clients.	Estimate client uncertainty using model predictions on public data.
Ouyang et al. [32]	2025	Clean	Weight clients.	Assign weights by optimising global model performance on clean public data.

information for client selection. Morafah et al. [14] test client models on a clean public dataset and rank them based on accuracy to identify noisy clients, which is used as a pruning criterion.

To reduce the dependency on clean labels, some research leverages model uncertainty. Lu et al. [31] use a public dataset to calculate prediction uncertainty of client models and optimise the global model without requiring labelled data. Ouyang et al. [32] also use a public dataset to assess client uncertainty without requiring labels; however, clean labels are still needed to evaluate model performance when adjusting aggregation weights.

In summary, these methods demonstrate the effectiveness of public dataset-based strategies for mitigating label noise in FL. Despite the importance of a public dataset at the server, most of these methods share the limitation that they require clean, labelled data, which suffers from challenges such as high labelling costs and vulnerability to performance degradation when the public dataset is noisy. As illustrated in Fig. 1, violations of this assumption can distort client reliability estimation and degrade global model performance, motivating the task of reliably estimating client reliability from an unlabelled and noisy public dataset, which is the focus of our work.

3. Method

3.1. Problem formulation

We consider an FLNL setting with J clients with unknown, heterogeneous label noise. Each client j holds a local dataset $\mathcal{D}_j$ and trains a local model $f_j(\cdot;\boldsymbol{\theta}_j)$ with parameters $\boldsymbol{\theta}_j$. In each communication round, clients upload their updated model parameters $\boldsymbol{\theta}_j$ to the central server, which aggregates them to obtain the global model parameters $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = \sum_{j=1}^{J} w^{(j)} \boldsymbol{\theta}_j, \tag{1}$$

where $w^{(j)}$ denotes the aggregation weight for client j. The aim of this paper is *to develop a theoretically well-founded and empirically superior method to estimate weights $w^{(j)}$ without requiring any labelled public dataset at the server for FLNL.*

Table 3 summarises the notation used in this section.

Table 3
Summary of notation.

Notation	Meaning
J	Number of clients
j	Index of a client ($j = 1, \ldots, J$)
$\mathcal{D}_j$	Local dataset held by client j
n_j	Number of samples in $\mathcal{D}_j$
$f_j(\cdot;\boldsymbol{\theta}_j)$	Local model of client j
$\boldsymbol{\theta}_j$	Local model parameters of client j
$\boldsymbol{\theta}$	Aggregated global model parameters
$w^{(j)}$	Aggregation weight for client j
C	Number of classes
$\mathcal{D}_{\text{pub}}$	Unlabelled public dataset at the server
N	Number of samples in the public dataset
x_i	i-th public data sample
T_i	Latent true label of sample x_i
$y_i^{(j)}$	Prediction of client j on sample x_i
$\mathbf{M}^{(j)}$	Confusion matrix of client j
$m_{c,l}^{(j)}$	Probability of predicting label l given true class c
$w_{DS}^{(j)}$	Reliability of client j
p_c	Prior probability of class c
$Q_i(c)$	Posterior probability that $T_i = c$
$\mathbf{1}\{\cdot\}$	Indicator function

3.2. Motivation

In FedAvg [33], the weights are proportional to the local dataset sizes:

$$w^{(j)} = \frac{n_j}{\sum_{l=1}^{J} n_l}, \tag{2}$$

where n_j is the number of local samples at client j. However, in practice, the quality of client data can vary significantly, especially when labels are corrupted by heterogeneous noise. Direct aggregation as in FedAvg [33] suffers from highly noisy clients, especially those with large sample sizes, to adversely affect the global model. To address this issue, it is natural to propose assigning aggregation weights based on each client's reliability in the target task. However, it is non-trivial to accurately estimate the reliability for each client, particularly as the level of label noise is unknown and heterogeneous across clients and there is only an unlabelled public dataset available in practice to facilitate such an estimation.

Let us first establish the notation for such a situation. Suppose the unlabelled public dataset held by the server is denoted by

$$\mathcal{D}_{\text{pub}} = \{x_1, x_2, \dots, x_N\}, \tag{3}$$

where the true label $T_i \in \{1, \dots, C\}$ for each sample x_i is unknown. Nevertheless, for each client j, the server can use the locally trained model f_j to make predictions on the public data, yielding the estimated label for sample x_i as

$$y_i^{(j)} \in \{1, \dots, C\}, \quad i = 1, \dots, N. \tag{4}$$

If ground-truth labels are known for the public dataset, we can summarise the predictive performance of model f_j for client j by calculating its confusion matrix:

$$\mathbf{M}^{(j)} = \left[m_{c,l}^{(j)}\right]_{c,l=1}^{C}, \tag{5}$$

where each entry of the confusion matrix is

$$m_{c,l}^{(j)} = P\left(y^{(j)} = l \mid T = c\right), \quad \sum_{l=1}^{C} m_{c,l}^{(j)} = 1, \tag{6}$$

which characterises the probability that client j's model f_j predicts label l given that the true label is c.

Once the confusion matrix is obtained, we can define the reliability of client j as the average of the diagonal entries of its confusion matrix:

$$w_{DS}^{(j)} = \frac{1}{C} \sum_{c=1}^{C} m_{c,c}^{(j)}. \tag{7}$$

This quantity measures the client j's mean accuracy across all classes.

However, since the public dataset is unlabelled, the ground-truth labels are unknown. Fortunately, as shown in Eqs. (6) and (7), we do not have to know the ground-truth labels, as we will only need the conditional probabilities $P(y^{(j)} = l \mid T = c)$ for each client j. To this end, we bring the Dawid-Skene model from statistical analysis to FL. The Dawid-Skene model enables us to first infer the posterior distribution of the true labels for each public sample by aggregating the predictions of all individual client models, and then the confusion matrix $\mathbf{M}^{(j)}$ can be estimated for each client. Finally, the estimated reliability scores $w_{DS}^{(j)}$ can be used to perform a reliability-aware weighted aggregation:

$$\boldsymbol{\theta} = \sum_{j=1}^{J} \frac{w_{DS}^{(j)}}{\sum_{l=1}^{J} w_{DS}^{(l)}} \boldsymbol{\theta}_j. \tag{8}$$

3.3. Dawid-skene-based estimation of client confusion matrices

Let us recall that $\mathbf{M}^{(j)} = [m_{c,l}^{(j)}]_{c,l=1}^{C}$ denotes the confusion matrix of client j (see Eq. (5)), and $w_{DS}^{(j)}$ denotes its reliability estimate computed from the diagonal average (see Eq. (7)). Given predictions $\{y_i^{(j)}\}_{i=1,j=1}^{N,J}$ of all clients on the public dataset $\mathcal{D}_{\text{pub}}$ at the server, the Dawid-Skene model can be used to jointly estimate the posterior distribution of true labels for each sample and the confusion matrices for all clients. Following the EM procedure, the Dawid-Skene algorithm alternates between estimating the posterior distribution of the latent true labels in E-step and updating the confusion matrices in M-step.

E-step. At the t-th iteration, for each sample x_i and each possible true class $c \in \{1, \dots, C\}$, the posterior probability of $T_i = c$, also called posterior responsibility, can be computed as

$$Q_i^{(t)}(c) = \frac{p_c^{(t)} \prod_{j=1}^{J} \left(m_{c,y_i^{(j)}}^{(j),(t)}\right)}{\sum_{c'=1}^{C} p_{c'}^{(t)} \prod_{j=1}^{J} \left(m_{c',y_i^{(j)}}^{(j),(t)}\right)}, \tag{9}$$

where $p_c^{(t)}$ is the current estimate of the prior probability of class c. It follows that $\sum_{c=1}^{C} Q_i^{(t)}(c) = 1$ for each sample.

M-step. The parameters (i.e., the confusion matrices and the class priors in Eq. (9)) are then updated to form the estimates for the next $(t+1)$-th iteration, as follows. For each client j and each (c,l) entry of its confusion matrix, we have

$$m_{c,l}^{(j),(t+1)} = \frac{\sum_{i=1}^{N} Q_i^{(t)}(c) \cdot \mathbf{1}\{y_i^{(j)} = l\}}{\sum_{l'=1}^{C} \sum_{i=1}^{N} Q_i^{(t)}(c) \cdot \mathbf{1}\{y_i^{(j)} = l'\}}, \tag{10}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function, and the class priors are updated as

$$p_c^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} Q_i^{(t)}(c). \tag{11}$$

The E-step and M-step are iterated until convergence.

After the convergence of the EM iterations, the updated reliability estimates $w_{DS}^{(j)}$ are computed as in Eq. (5) from the estimated confusion matrices and then applied in the global aggregation step described in Eq. (8). This ensures that clients with lower reliability have reduced influence on the global model, mitigating the impact of heterogeneous label noise.

4. Theoretical properties

In this section, we provide theoretical guarantees of FedDS in the FL setting, including convergence and identifiability. The notation is consistent with that in Section 3. We note that, given the true label T_i of a public sample, the predictions $\{y_i^{(j)}\}_{j=1}^{J}$ from different clients are assumed to be conditionally independent, as assumed in the EM algorithm for the Dawid-Skene model for Eq. (9); and each client's confusion matrix remains fixed during one EM estimation process.

4.1. Convergence

Under the Dawid-Skene model, the marginal probability of the observed predictions for a public sample x_i is given by

$$P\left(y_i^{(1)}, \ldots, y_i^{(J)}\right) = \sum_{c=1}^{C} p_c \prod_{j=1}^{J} \left(m_{c,y_i^{(j)}}^{(j)}\right). \tag{12}$$

The incomplete-data likelihood for all observed predictions is

$$L\left(\{\mathbf{M}^{(j)}\}, \{p_c\}; \{y_i^{(j)}\}\right) = \prod_{i=1}^{N} \left(\sum_{c=1}^{C} p_c \prod_{j=1}^{J} \left(m_{c,y_i^{(j)}}^{(j)}\right)\right), \tag{13}$$

and the corresponding log-likelihood is

$$\ell\left(\{\mathbf{M}^{(j)}\}, \{p_c\}; \{y_i^{(j)}\}\right) = \sum_{i=1}^{N} \log\left(\sum_{c=1}^{C} p_c \prod_{j=1}^{J} \left(m_{c,y_i^{(j)}}^{(j)}\right)\right). \tag{14}$$

The E-step and M-step of the EM algorithm update the posterior responsibilities, confusion matrices, and class priors at each iteration t as defined in Eqs. (9)–(11), respectively. The EM convergence property applies [34], guaranteeing that the log-likelihood does not decrease at each iteration:

$$\ell\left(\{\mathbf{M}^{(j),(t+1)}\}, \{p_c^{(t+1)}\}; \{y_i^{(j)}\}\right) \geq \ell\left(\{\mathbf{M}^{(j),(t)}\}, \{p_c^{(t)}\}; \{y_i^{(j)}\}\right). \tag{15}$$

Since the log-likelihood function is bounded from above, the above non-decreasing sequence must converge. Consequently, the sequence of parameter estimates $\{\mathbf{M}^{(j),(t)}, p_c^{(t)}\}$ generated by the EM algorithm converges to a stationary point of the likelihood function, which may correspond to a local maximum or a saddle point.

Therefore, under the Dawid-Skene model and the conditional independence assumption of the client predictions, the EM estimation procedure adopted in this work is convergent. This provides a theoretical guarantee for the stability of the proposed confusion matrix estimation.

4.2. Identifiability

In general, the Dawid-Skene model is not strictly identifiable. In a latent class model, a simultaneous relabelling of the latent class index, together with a corresponding permutation of the rows of all confusion matrices and the class prior, leaves the joint distribution of the observed predictions unchanged [16]. As a result, without additional constraints, there is in general no unique parameter set $\{\mathbf{M}^{(j)}, p_c\}$ that corresponds to a given joint distribution of observed labels $\{y_i^{(j)}\}$.

Nevertheless, among the true-label-permuted estimates that correspond to the same joint distribution of observed labels $\{y_i^{(j)}\}$, it is sensible to assume that the client confusion matrices that are diagonally dominant are more likely to be the correct ones [16]. By diagonal dominance, we mean that for every class c the confusion matrix of client j satisfies

$$m_{c,c}^{(j)} > m_{c,l}^{(j)}, \quad \forall\, l \neq c. \tag{16}$$

Here we shall show that, if the ground-truth confusion matrices of clients are truly diagonally dominant, these matrices can be uniquely identified from the joint distribution of the observed predictions.

We first establish the notation. Let $\{\mathbf{M}^{(j)}, p_c\}$ denote the true parameter set. We assume that there exists at least one client whose confusion matrix is diagonally dominant. Assume client j is such a client satisfying Eq. (16). Consider any permutation σ of the class index set $\{1, \ldots, C\}$, and the corresponding permuted parameter set $\{\widetilde{\mathbf{M}}^{(j)}, \widetilde{p}_c\}$ denoted by

$$\widetilde{p}_c = p_{\sigma(c)}, \quad \widetilde{m}^{(j)}_{c,l} = m^{(j)}_{\sigma(c),l}, \quad \forall\, c, l, j. \tag{17}$$

Recall that this permuted parameter set induces the same joint distribution of observed labels $\{y_i^{(j)}\}$ as does the original one.

Then we show that, however, the permuted confusion matrix $\widetilde{\mathbf{M}}^{(j)}$ cannot also be diagonally dominant for any non-trivial permutation σ, where there exists at least one class c such that $\sigma(c) \neq c$. Recall from Eq. (17) that the c-th row of $\widetilde{\mathbf{M}}^{(j)}$ is obtained from the $\sigma(c)$-th row of $\mathbf{M}^{(j)}$:

$$\widetilde{m}^{(j)}_{c,l} = m^{(j)}_{\sigma(c),l}, \quad \forall\, l. \tag{18}$$

It follows that, when $l = \sigma(c)$,

$$\widetilde{m}^{(j)}_{c,\sigma(c)} = m^{(j)}_{\sigma(c),\sigma(c)}. \tag{19}$$

In the meantime, by the diagonal dominance assumption in Eq. (16) for $\mathbf{M}^{(j)}$, we have

$$m^{(j)}_{\sigma(c),\sigma(c)} > m^{(j)}_{\sigma(c),l}, \quad \forall\, l \neq \sigma(c). \tag{20}$$

Hence, as $c \neq \sigma(c)$, it follows Eq. (19), then Eq. (20) and finally Eq. (18), that

$$\widetilde{m}^{(j)}_{c,\sigma(c)} = m^{(j)}_{\sigma(c),\sigma(c)} > m^{(j)}_{\sigma(c),c} = \widetilde{m}^{(j)}_{c,c}. \tag{21}$$

That is, the largest entry in the c-th row of $\widetilde{\mathbf{M}}^{(j)}$ is not on the diagonal, or say, $\widetilde{\mathbf{M}}^{(j)}$ is not diagonally dominant.

In short, under the joint estimation within the Dawid-Skene framework and the diagonal dominance assumption valid for at least one client, the client confusion matrices can be uniquely determined by the observed joint distribution of the client predictions, and hence are identifiable.

5. Experiments

5.1. Experimental setup

Datasets. We use three standard benchmark datasets in our experiments: MNIST [35], CIFAR-10 [36], and CIFAR-100 [36]. MNIST contains 60,000 training images and 10,000 test images with ten categories. CIFAR-10 and CIFAR-100 both consist of 50,000 training images and 10,000 test images with ten and one hundred categories, respectively.

In order to simulate the FL setting, we partition the dataset into three subsets. First, 10% of the original training data are held out as a public dataset that is accessible to the server. This split is performed by using stratified sampling, ensuring that the public dataset covers all classes present in the full training set while preserving the original class proportions. Second, the remaining 90% of the training data are divided among a fixed number of clients. Finally, the original test dataset is retained at the server for global model evaluation. We report the test accuracy of the global model, which is calculated as the average accuracy over the final 10 communication rounds of each experiment. To ensure a fair comparison, all methods evaluated in our experiments use the same data partition that includes a public dataset.

Moreover, to investigate the impact of data heterogeneity, we design two data partitions to represent the independent and identically distributed (IID) scenario and the non-IID scenario. First, in the IID scenario, the training data is randomly shuffled and partitioned equally among all clients. Second, in the more challenging non-IID scenario, we employ a Dirichlet distribution to assign different proportions of class labels to each client [37]. We control the degree of heterogeneity by tuning the hyperparameter of the Dirichlet distribution α. Specifically, we set $\alpha = 0.5$ to simulate a highly heterogeneous scenario where each client likely possesses samples from only a few classes and $\alpha = 10$ to simulate a moderately heterogeneous scenario.

In addition, to simulate heterogeneous clients with varying data quality, we introduce symmetric label noise to each client's local training set. For a client with an assigned noise rate ρ, each data label has a probability of $1 - \rho$ of being correct and a probability of ρ of being uniformly flipped to a random incorrect class. To model a diverse range of client reliability, the noise rates are drawn from a discrete uniform distribution ranging from 0.1 to 1.0 with a step size of 0.1 across the clients.

Implementation details. Following [23,32,33], we use distinct model architectures for different datasets. For MNIST, we employ a simple convolutional neural network (CNN). For CIFAR-10, we use a VGG-style CNN [38]. For the more complex CIFAR-100, we adopt a ResNet-18 architecture [39]. In the experiments, we consider a total of 100 communication rounds. Following the vanilla setting of FL [33], the system comprises 100 clients, of which 10 clients are randomly selected in each round to perform local updates. Each selected client is trained for 5 local epochs with a batch size of 64. We adopt stochastic gradient descent (SGD) as the optimization method, with a learning rate of 0.01 and momentum of 0.9. All experiments are implemented in PyTorch and run on an NVIDIA Tesla P100 GPU under a Linux-based environment.

5.2. Plug-and-play enhancement of FL methods originally not addressing label noise

To validate the effectiveness of our proposed FedDS as a plug-and-play module to boost the performance of mainstream FL methods that do not address label noise, we integrate FedDS with FedAvg [33], FedProx [40], SCAFFOLD [41] and MOON [42]. In practice,

Table 4
Test accuracy (%) comparison between several FL methods in their vanilla form and their FedDS-enhanced versions. The Δ row quantifies the absolute performance improvement provided by our module across all datasets and data distributions. Values in grey are the FedDS-enhanced results, and values in green are the corresponding improvements achieved by FedDS.

Baseline		IID			Non-IID ($\alpha = 0.5$)			Non-IID ($\alpha = 10$)		
		MNIST	CIFAR-10	CIFAR-100	MNIST	CIFAR-10	CIFAR-100	MNIST	CIFAR-10	CIFAR-100
FedAvg [33]	Vanilla	95.57	55.15	22.45	91.73	37.24	20.08	94.04	40.42	21.71
	+FedDS	96.98	59.68	31.54	94.96	43.02	28.03	96.34	50.98	30.43
	Δ	+1.41	+4.53	+9.09	+3.23	+5.78	+7.95	+2.30	+10.56	+8.72
FedProx [40]	Vanilla	95.79	55.78	23.87	93.52	38.01	21.82	95.64	44.27	22.69
	+FedDS	97.17	60.02	29.61	95.01	43.14	26.42	96.41	51.64	28.95
	Δ	+1.38	+4.24	+5.74	+1.49	+5.13	+4.60	+0.77	+7.37	+6.26
SCAFFOLD [41]	Vanilla	92.70	53.45	22.96	91.18	36.75	20.14	91.59	42.21	22.23
	+FedDS	95.78	60.55	34.64	94.17	43.05	28.56	94.86	51.06	33.02
	Δ	+3.08	+7.10	+11.68	+2.99	+6.30	+8.42	+3.27	+8.85	+10.79
MOON [42]	Vanilla	95.28	36.32	27.39	80.42	27.96	23.56	95.09	33.19	26.34
	+FedDS	96.51	48.39	35.51	92.23	32.69	27.79	96.35	43.87	34.06
	Δ	+1.23	+12.07	+8.12	+11.81	+4.73	+4.23	+1.26	+10.68	+7.72

Table 5
Test accuracy (%) comparison of all methods across different datasets and data distribution settings. $\alpha = 0.5$: a highly heterogeneous scenario; $\alpha = 10$: a moderately heterogeneous scenario. The best is in bold, and the second best is underlined. Values in grey are the results achieved by the proposed FedDS.

Baseline	IID			Non-IID ($\alpha = 0.5$)			Non-IID ($\alpha = 10$)		
	MNIST	CIFAR-10	CIFAR-100	MNIST	CIFAR-10	CIFAR-100	MNIST	CIFAR-10	CIFAR-100
FedAvg [33]	95.57	55.15	22.45	91.73	37.24	20.08	94.04	40.42	21.71
FOCUS [13]	96.24	<u>59.39</u>	**32.51**	<u>92.47</u>	33.07	<u>27.58</u>	95.84	43.33	**31.83**
Client Selection [15]	**97.00**	50.59	24.28	92.06	**44.65**	21.90	**96.72**	**51.46**	23.40
ClipFL [14]	95.70	54.42	22.27	83.58	29.99	19.74	94.83	38.97	21.31
FedDS	<u>96.98</u>	**59.68**	<u>31.54</u>	**94.96**	<u>43.02</u>	**28.03**	<u>96.34</u>	<u>50.98</u>	<u>30.43</u>

this requires no modification to the local training process of existing FL methods, as FedDS is inserted only at the aggregation stage, where it replaces the original aggregation weights with the weights estimated from our module. For each method, we compare its original implementation against an enhanced version where our method is embedded. All methods are evaluated under identical experimental configurations as shown in Section 5.1 to ensure a fair comparison.

As shown in Table 4, our FedDS method can clearly boost the accuracies of FL methods under noisy labels, as the Δ rows confirm a consistent performance boost for each FL method in every setting we test, showing the high compatibility of FedDS. The boost in performance is especially large for SCAFFOLD [41] and MOON [42], which were severely affected by label noise. Integrating our FedDS into these methods can increase their accuracy in all cases, often significantly by more than 7–12%. This can be attributed to the fact that FedDS can act as a noise estimator during the aggregation stage, allowing FL methods to work on a more reliable set of updates. Therefore, our method can serve as a valuable enhancement module that gives FL methods the robustness needed to work with the noisy labels commonly seen in real-world applications.

5.3. Comparison with SOTA FLNL methods

To further validate the effectiveness of our proposed Dawid-Skene-based weighted aggregation strategy, we also conduct a comparison with several SOTA FLNL methods that also leverage a public dataset. Our comparison includes the standard FedAvg [33] as a baseline, which performs equal-weight averaging. We also compare against ClipFL [14], a client pruning method that removes underperforming clients in the training process, and FOCUS [13], a client re-weighting algorithm that calculates client weights based on a combination of the local model's loss on the public dataset and the global model's loss on the client's local data. In our comparative experiments here, all baseline methods except FedAvg [33] use clean, labelled public datasets containing no noisy labels, whereas our approach uses an unlabelled public dataset. All methods are evaluated under identical experimental configurations as shown in Section 5.1. Because the compared methods [13–15] rely on clean, labelled public datasets, the aim of this experiment is to examine whether FedDS can achieve comparable performance to them when the requirement for clean, labelled public data at the server is relaxed.

The experimental results presented in Table 5 validate the competitive performance of our proposed FedDS method across a diverse range of datasets and data distributions. Among the nine scenarios listed in Table 5, FedDS achieves the highest accuracy in three cases: on CIFAR-10 in the IID setting, and on MNIST and CIFAR-100 in the highly heterogeneous settings (Non-IID, $\alpha = 0.5$), which indicates its robustness under severe client heterogeneity. Moreover, we note that, in the other six settings including the moderately heterogeneous settings (Non-IID, $\alpha = 10$), our FedDS is still competitive, being the second best performer and within only a small margin from the best method. It is also important to note that these competitive results of FedDS are achieved by using only an unlabelled public dataset, whereas the other competing FLNL methods rely on clean, labelled public datasets to perform.

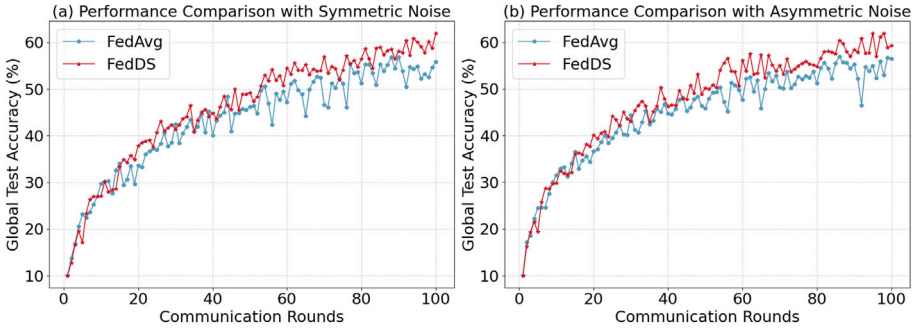

Fig. 3. Demonstrating the robustness of our proposed method to different types of label noise. The figure compares the test accuracy of our proposed method against FedAvg on CIFAR-10 with (a) symmetric noise and (b) the more challenging asymmetric noise.

Nonetheless, we note that, on more complex CIFAR-10 and CIFAR-100 datasets than on simpler MNIST, all the accuracies of FedDS and the compared FLNL methods remain much lower, as shown in Table 5. This pattern reflects the inherent difficulty of FL under severe data heterogeneity, data scarcity, and label noise. Compared with MNIST, CIFAR-10 and CIFAR-100 have much higher visual complexity and intra-class variation. Combined with heterogeneous and often high levels of label noise across clients and limited local data for training, noisy gradients may heavily affect local training, limiting the amount of reliable discriminative information that can be learned and the performance of subsequent aggregation of local models.

In summary, these results indicate that FedDS can work effectively under weaker and more realistic assumptions of the availability of public data annotations, and equally importantly, that FedDS can offer a robust and practical solution to FL under label noise. That said, there is still substantial room for FedDS and other FLNL methods to improve their performance on complex datasets.

5.4. Ablation studies

5.4.1. Effect of type of noise

The aim of this experiment is to assess whether the proposed method remains effective under different label noise structures. We therefore consider both symmetric and asymmetric noise settings, and adopt FedAvg [33] as a baseline. Symmetric noise refers to labels being randomly flipped to any other incorrect class, while asymmetric noise involves flipping to specific, confusable classes, posing a greater challenge to model training. For our experiments on the CIFAR-10 dataset, we generate asymmetric noise by targeting four pairs of semantically similar classes. Specifically, a certain percentage of labels are systematically flipped (Truck labels are changed to Automobile, Bird labels to Airplane, Deer labels to Horse, and Cat labels to Dog). For each of these source classes, we flip a proportion ρ of its samples to the corresponding target class, where ρ denotes that client's assigned noise rate introduced earlier. This type of structured noise poses a greater challenge to model training, as the mislabelling is not random but correlated with the data's features, potentially causing the model to learn incorrect patterns.

The resulting accuracy curves, shown in Fig. 3, clearly demonstrate that our proposed method consistently outperforms the baseline in both environments. Under symmetric noise, the performance advantage of our proposed method widens as training progresses, achieving an average accuracy of approximately 59.39% over the last 10 rounds, significantly higher than FedAvg's 55.15%. In the more challenging asymmetric noise environment, our proposed method maintains its lead, with its average accuracy over the final 10 rounds reaching 58.65%, surpassing FedAvg's 53.77%. These results demonstrate the robustness of our method against various noise structures: Whether the errors are random (symmetric) or systematic (asymmetric), our proposed method can effectively identify and down-weight the contributions of unreliable clients, leading to a more accurate global model.

5.4.2. Effect of size of public dataset

To examine the impact of public dataset size on the performance of FedDS, we extract samples from CIFAR-10 at three proportions, 0.05, 0.1 and 0.2, respectively to construct three public datasets of different sizes, while keeping the training set size for clients unchanged at 40,000 samples so that the factor of public dataset size can be isolated during the examination.

The performance of these three public datasets is plotted in Fig. 4, which shows that the performance improves as the size of the public dataset increases: The average test accuracy increases from 37.31% at proportion of 0.05, to 37.94% at proportion of 0.1, and further to 38.90% at proportion of 0.2. This trend implies that a larger public dataset provides more information for estimating the confusion matrices, leading to more accurate reliability estimation. In practice, using a larger public dataset nonetheless means higher computational overhead, since each client needs to make predictions on all public samples in every communication round. Therefore, considering the trade-off between the performance gain and the computational overhead, we set the proportion to 0.1 as a relatively balanced choice.

5.4.3. Effect of type of public dataset

The objective of this experiment is to investigate how the distributional characteristics of the public dataset, used for the Dawid-Skene-based aggregation, influence the performance of the global model. The client data consists of the CIFAR-10 training dataset with heterogeneous label noise, and the global task is the CIFAR-10 classification. With the client data and the global task fixed, the

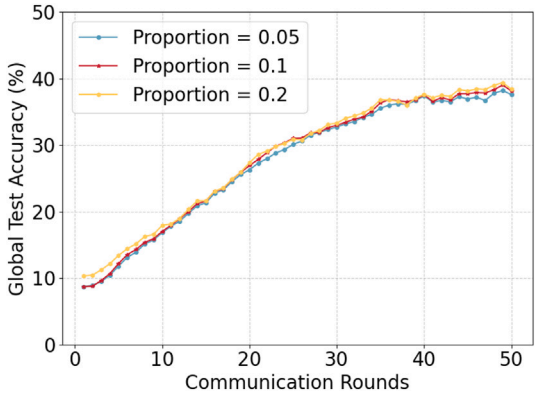

Fig. 4. Effect of public dataset size on global test accuracy, where the public datasets used for Dawid-Skene-Model-based aggregation are constructed at sampling proportions of 0.05, 0.1 and 0.2, respectively, while the training set size is fixed for clients at 40,000 samples.

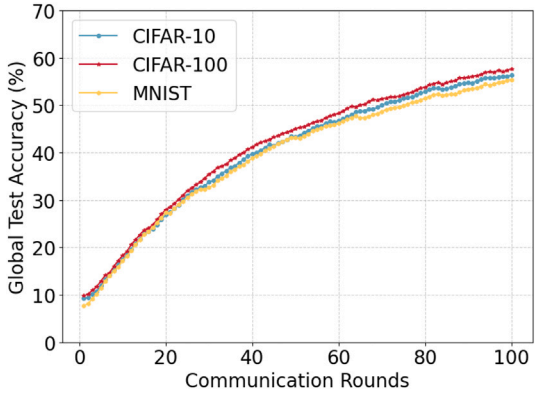

Fig. 5. Effect of type of public dataset on global test accuracy, comparing CIFAR-10, CIFAR-100 and MNIST as the public dataset while keeping the client dataset and the global task fixed.

performance of the global model is evaluated across three scenarios with distinct public datasets: CIFAR-10 which represents data with an identical domain and class distribution to the primary task; CIFAR-100 which uses data that are out of distribution regarding classes yet retains domain similarity; and MNIST which introduces a domain shift concerning visual features.

The results in Fig. 5 show the stability of performance when the public dataset maintains domain similarity to the client data. The final mean accuracy is 56.1% for CIFAR-10 and is 59.0% for CIFAR-100, suggesting that the Dawid-Skene model's capacity to identify reliable clients is rather robust to differences in the label set of the public dataset, provided that the underlying visual domains are consistent. The higher accuracy with CIFAR-100 may be due to its larger number of fine-grained categories offering more information when estimating client reliability for the simpler target task of CIFAR-10 classification. In contrast, when MNIST is used as the public dataset, the performance drops to 54.5%, which can be attributed to the significant domain shift from CIFAR-10. This observation implies that FedDS prefers the unlabelled public dataset to be domain-relevant for precise estimation of client reliability.

In conclusion, the experiment demonstrates that the Dawid-Skene-based aggregation method remains effective when the public data come from a domain the same as or relevant to the target task, even if the class distributions are not identical.

5.4.4. Effect of number of iterations

This experiment aims to explore the impact of the number of iterations in the Dawid-Skene algorithm on the aggregation effect of the FedDS. Keeping other FL and local training parameters unchanged, we only change the iteration number in the FedDS algorithm, with four values of 50, 100, 500 and 1000 evaluated, respectively.

As shown in Fig. 6(a), the number of iterations in the FedDS algorithm can affect final performance. Poor performance is observed with a small number of iterations (iter = 50), achieving only 56.06% mean accuracy, suggesting that the FedDS algorithm requires sufficient iterations to converge to the clients' true reliability. The performance of FedDS improves with iterations, achieving 60.46% mean accuracy at iter = 500. However, many more iterations (iter = 1000) do not yield any performance gain, but instead result in a slight drop to 58.98%. This observation may be attributed to overfitting to local label noise under limited public data by the EM procedure in the Dawid-Skene algorithm [43,44].

To further examine this observation, we conduct an additional experiment with a larger public dataset at public proportion of 0.2. In this setting, FedDS with 1000 iterations outperforms FedDS with 500 iterations, as shown in Fig. 6(b). That is, with a larger

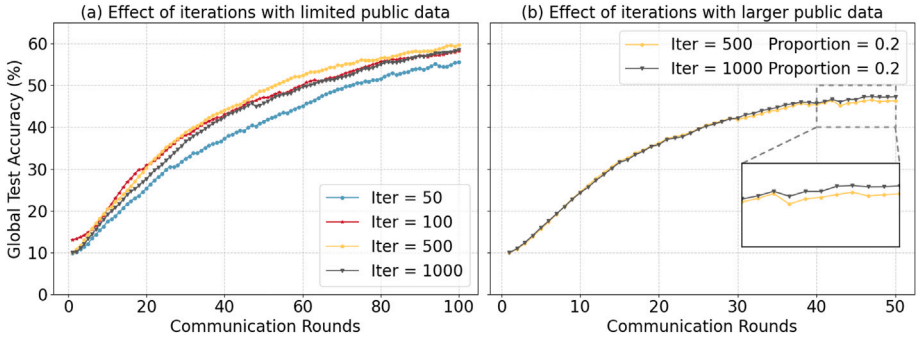

Fig. 6. Effect of the number of FedDS iterations for client reliability estimation on global test accuracy. (a) Public proportion = 0.1, where performance is evaluated under 50, 100, 500 and 1000 iterations. (b) Public proportion = 0.2, comparing 500 and 1000 iterations.

public dataset, more iterations can be preferred. However, we note that in such a case we should also consider a trade-off between performance gain and computational overhead while making a balanced choice of iteration number.

5.5. Computational complexity and practical overhead

In each communication round, the dominant computational cost remains for the local model training at the client side, the same as in FedAvg [33]. The additional computation introduced by FedDS occurs only at the server due to the Dawid-Skene procedure on a small public dataset to estimate posterior label probabilities and confusion matrices of clients. The computational cost of this process grows with the number of participating clients, the size of the public dataset, the number of classes, and the number of EM iterations. In practice, the public dataset is small, the number of classes is limited, the number of EM iterations is moderate, and the Dawid-Skene procedure involves only simple operations without requiring gradient computation. As a result, the additional server-side overhead introduced by FedDS is light and does not affect the scalability.

5.6. Summary of experimental results

The experimental results demonstrate the effectiveness and robustness of FedDS for FL, especially under highly heterogeneous label noise. The plug-and-play experiments show that FedDS can be easily integrated into mainstream FL methods, and across multiple datasets and data distributions, FedDS can consistently improve their performance. Comparison with state-of-the-art FLNL methods indicates that FedDS achieves competitive performance despite relying on only weaker assumptions of public data, with clearer advantages shown in highly heterogeneous non-IID settings. Additional ablation studies further confirm that the performance of FedDS is stable under different noise types and reasonable hyperparameter choices. Nevertheless, compared with that on MNIST, the performance on CIFAR-10 and CIFAR-100 remains lower for FedDS and other competing FLNL methods, due to the increased difficulty of FL under severe label noise, client heterogeneity, and limited local data.

6. Conclusion

In this paper, to address the challenge of heterogeneous label noise in FLNL, we introduce FedDS, a novel method that brings the Dawid-Skene model to FL. FedDS can effectively estimate the reliability of each client under heterogeneous label noise and weight clients based on their reliability estimates, without requiring any labelled public dataset at the server. Extensive experimental results showcase the competitive performance of FedDS. We note that the performance of FedDS and other FLNL methods on complex datasets remains low, and the main contribution of FedDS is in enhancing robustness and practicality of FLNL under weaker and more realistic requirements for only an unlabelled dataset available at the server.

While FedDS significantly improves practicality by removing the requirement for clean, labelled data at the server, a natural direction for future work is to develop a method that can eliminate the need for any public dataset, labelled or unlabelled, for broader and stronger generalisation. This could be achieved by, for example, developing self-supervised methods for estimating client reliability locally at the client side, or exploring generative techniques to synthesise clean data at the server to support robust aggregation.

CRediT authorship contribution statement

Jia Dong: Writing – original draft, Visualization, Software, Resources, Methodology, Investigation, Formal analysis, Data curation. **Rui Zhu:** Writing – review & editing, Validation, Supervision. **Xinyi Shang:** Writing – review & editing, Validation. **Jing-Hao Xue:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

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.

Data availability

Public benchmark datasets are used, and the code is publicly available at https://github.com/Gia99999/FedDS.

References

[1] W. Huang, M. Ye, Z. Shi, G. Wan, H. Li, B. Du, Q. Yang, Federated learning for generalization, robustness, fairness: a survey and benchmark, IEEE Trans. Pattern Anal. Mach. Intell. 46 (12) (2024) 9387–9406.
[2] R. Al-Huthaifi, T. Li, W. Huang, J. Gu, C. Li, Federated Learning in smart Cities: privacy and security Survey, Inf. Sci. 632 (2023) 833–857.
[3] P. Qi, D. Chiaro, A. Guzzo, M. Ianni, G. Fortino, F. Piccialli, Model aggregation techniques in federated Learning: a comprehensive survey, Futur. Gener. Comput. Syst. 150 (2024) 272–293.
[4] M. Arbaoui, M.-E.-A. Brahmia, A. Rahmoun, M. Zghal, Federated Learning Survey: a multi-level taxonomy of aggregation techniques, experimental insights, and future frontiers, ACM Trans. Intell. Syst. Technol. 15 (6) (2024) 1–69.
[5] J. Li, T. Chen, S. Teng, A comprehensive survey on client selection strategies in federated Learning, Comput. Netw. 251 (2024) 110663.
[6] S. Liang, J. Huang, J. Hong, D. Zeng, J. Zhou, Z. Xu, Fednoisy: federated noisy label learning benchmark, arXiv preprint arXiv:2306.11650, 2025.
[7] X. Jiang, J. Li, N. Wu, Z. Wu, X. Li, S. Sun, G. Xu, Y. Wang, Q. Li, M. Liu, Fnbench: benchmarking robust federated learning against noisy labels, arXiv preprint arXiv:2505.06684, 2025.
[8] J. Pei, W. Liu, J. Li, L. Wang, C. Liu, A review of federated learning methods in heterogeneous scenarios, IEEE Trans. Consum. Electron. 70 (3) (2024) 5983–5999.
[9] A. Mora, A. Bujari, P. Bellavista, Enhancing generalization in federated learning with heterogeneous data: a comparative literature review, Futur. Gener. Comput. Syst. 157 (2024) 1–15.
[10] X. Ji, Z. Zhu, W. Xi, O. Gadyatskaya, Z. Song, Y. Cai, Y. Liu, Fedfixer: mitigating heterogeneous label noise in federated learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, 2024, pp. 12830–12838.
[11] B. Zeng, X. Yang, Y. Chen, Z. Shen, H. Yu, Y. Zhang, Fedes: federated early-stopping for hindering memorizing heterogeneous label noise, in: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, 2024, pp. 5416–5424.
[12] J. Dong, R. Zhu, X. Shang, J.-H. Xue, Federated learning with noisy labels: a comprehensive and concise review of current methodologies and future directions, Neural Netw. (2026) (accepted).
[13] Y. Chen, X. Yang, X. Qin, H. Yu, P. Chan, Z. Shen, Dealing with label quality disparity in federated learning, in: Federated Learning: Privacy and Incentive, Springer, 2020, pp. 108–121.
[14] M. Morafah, H. Chang, C. Chen, B. Lin, Federated learning client pruning for noisy labels, ACM Trans. Model. Perform. Eval. Comput. Syst. 10 (2) (2025) 1–25.
[15] M. Yang, H. Qian, X. Wang, Y. Zhou, H. Zhu, Client selection for federated learning with label noise, IEEE Trans. Veh. Technol. 71 (2) (2021) 2193–2197.
[16] A.P. Dawid, A.M. Skene, Maximum likelihood estimation of observer error-rates using the em algorithm, J. R. Stat. Soc. Ser. C Appl. Stat. 28 (1) (1979) 20–28.
[17] W. Han, X. Yan, A privacy preserving federated learning aggregation algorithm for noise label, in: 2023 9th International Conference on Computer and Communications (ICCC), IEEE, 2023, pp. 2170–2176.
[18] W. Sun, R. Yan, R. Jin, R. Zhao, Z. Chen, Curriculum-based federated learning for machine fault diagnosis with noisy labels, IEEE Trans. Ind. Inf. 20 (12) (2024) 13820–13830.
[19] J. Xu, Z. Chen, T.Q.S. Quek, K.F.E. Chong, Fedcorr: multi-stage federated learning for label noise correction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10184–10193.
[20] B. Zeng, X. Yang, Y. Chen, H. Yu, C. Hu, Y. Zhang, Federated data quality assessment approach: robust learning with mixed label noise, IEEE Trans. Neural Netw. Learn. Syst. 35 (12) (2024) 17620–17634.
[21] J. Li, G. Li, H. Cheng, Z. Liao, Y. Yu, Feddiv: collaborative noise filtering for federated learning with noisy labels, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, 2024, pp. 3118–3126.
[22] T.-T. Giap, T.-D. Kieu, T.-L. Le, T.-H. Tran, Feddc: label noise correction with dynamic clients for federated learning, IEEE Internet Things J. 12 (8) (2025) 10266–10277.
[23] Y. Xu, Y. Liao, L. Wang, H. Xu, Z. Jiang, W. Zhang, Overcoming noisy labels and non-iid data in edge federated learning, IEEE Trans. Mob. Comput. 23 (12) (2024) 11406–11421.
[24] V. Tsouvalas, A. Saeed, T. Ozcelebi, N. Meratnia, Labeling chaos to learning harmony: federated learning with noisy labels, ACM Trans. Intell. Syst. Technol. 15 (2) (2024) 1–26.
[25] S. Yu, J.-H. Ahn, J. Kang, Fedefc: federated learning using enhanced forward correction against noisy labels, arXiv preprint arXiv:2504.05615, 2025.
[26] R. Pu, L. Yu, S. Zhan, G. Xu, F. Zhou, C.X. Ling, B. Wang, Fedelr: when federated learning meets learning with noisy labels, Neural Netw. 187 (2025) 107275.
[27] G.F. Ejigu, A. Adhikary, C.S. Hong, Relaxed contrastive learning for robust federated models with noisy labels and limited clients, in: 2025 27th International Conference on Advanced Communications Technology (ICACT), IEEE, 2025, pp. 1–6.
[28] G.F. Ejigu, K. Kim, C.S. Hong, Mitigating label noise in federated learning with regularized features and robust loss, IEEE Trans. Artif. Intell. (2025), https://doi.org/10.1109/TAI.2025.3609745
[29] T. Tuor, S. Wang, B.J. Ko, C. Liu, K.K. Leung, Overcoming noisy and irrelevant data in federated learning, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 5020–5027.
[30] W. Wu, L. He, W. Lin, C. Maple, Fedprof: selective federated learning based on distributional representation profiling, IEEE Trans. Parallel Distrib. Syst. 34 (6) (2023) 1942–1953.
[31] Y. Lu, L. Chen, Y. Zhang, Y. Zhang, B. Han, Y.-M. Cheung, H. Wang, Federated learning with extremely noisy clients via negative distillation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, 2024, pp. 14184–14192.
[32] C. Ouyang, J. Mao, Y. Li, T. Li, D. Zhu, C. Zhou, Z. Xu, Federated learning for extreme label noise: enhanced knowledge distillation and particle swarm optimization, Electronics 14 (2) (2025) 366.
[33] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. Arcas y, Communication-Efficient learning of deep networks from decentralized data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, vol. 54, PMLR, 2017, pp. 1273–1282.
[34] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. R. Stat. Soc. Ser. B (Methodol.) 39 (1) (1977) 1–22.
[35] Y. LeCun, The Mnist database of handwritten digits, 1998. http://yann.lecun.com/exdb/mnist/.
[36] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Tech. rep., University of Toronto, Toronto, Ontario, 2009.
[37] T.-M.H. Hsu, H. Qi, M. Brown, Measuring the effects of non-identical data distribution for federated visual classification, arXiv preprint arXiv:1909.06335, 2019.
[38] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, 2014.
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[40] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proc. Mach. Learn. Syst. 2 (2020) 429–450.
[41] S.P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, A.T. Suresh, Scaffold: stochastic controlled averaging for federated learning, in: International Conference on Machine Learning, PMLR, 2020, pp. 5132–5143.
[42] Q. Li, B. He, D. Song, Model-contrastive federated learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10713–10722.
[43] C. Liu, Y.-M. Wang, Truelabel + confusions: a spectrum of probabilistic models in analyzing multiple ratings, in: Proceedings of the 29th International Coference on International Conference on Machine Learning, 2012, pp. 17–24.
[44] Y. Ye, T. Wang, M. Zhang, D. Feng, Revisiting EM-based estimation for locally differentially private protocols, in: Proceedings 2025 Network and Distributed System Security Symposium, 2025.