EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems
Research Article EAI.EU

FedNDA: Enhancing Federated Learning with Noisy Client Detection and Robust Aggregation

Tuan-Dung Kieu[1,2], Charles Fonbonne[3], Trung-Kien Tran[4], Thi Lan Le[5], Hai Vu[5], Huu Thanh Nguyen[5], Thanh-Hai Tran[5,*]

[1]School of Information and Communications Technology, Hanoi University of Science and Technology, Hanoi, Vietnam
[2]Thuyloi University, Hanoi, Vietnam
[3]Toulon University, France
[4]Institute of Information Technology, AMST, Hanoi, Vietnam
[5]School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam

Abstract

Federated Learning is a novel decentralized methodology that enables multiple clients to collaboratively train a global model while preserving the privacy of their local data. Although federated learning enhances data privacy, it faces challenges related to data quality and client behavior. A fundamental issue is the presence of noisy labels in certain clients, which damages the global model's performance. To address this problem, this paper introduces a **Fed**erated learning framework with **N**oisy client **D**etection and robust **A**ggregation, FedNDA. In the first stage, FedNDA detects noisy clients by analyzing the distribution of their local losses. A noisy client exhibits a loss distribution distinct from that of clean clients. To handle the class imbalance issue in local data, we utilize per-class losses instead of the total loss. We then assign each client a noisiness score, calculated as the Earth Mover's Distance between the per-class loss distribution of the client and the average distribution of all clean clients. This noisiness metric is more sensitive for detecting noisy clients compared to conventional metrics such as Euclidean distance or L_1 norm. The noisiness score is subsequently transferred to and used in the server-side aggregation function to prioritize clean clients while reducing the influence of noisy clients. Experimental results demonstrate that FedNDA consistently outperforms state-of-the-art methods such as FedAvg, FedNoRo, FedCorr, and FedELC on two benchmark datasets, CIFAR-10 and ICH. Notably, FedNDA achieves the highest accuracy in both clean and noisy client scenarios, maintaining robust performance regardless of optimizer or preprocessing strategy. Our code is available at: https://github.com/ktzung/FedNDA.

Received on 16 February 2025; accepted on 21 June 2025; published on 03 July 2025

Keywords: Federated Learning; Deep Learning; Noisy Client; Non-IID; Class-imbalance

doi:10.4108/eetinis.v12i3.8720

1. INTRODUCTION

Federated Learning (FL) is a decentralized approach for training machine learning models in which data remain on local devices, and only model updates are shared with a central server [1], [2], [3]. This ensures data privacy, making FL suitable for applications in sensitive domains such as healthcare and finance. It also enables collaboration across diverse data sources while reducing communication and storage costs. However, it faces challenges such as data heterogeneity, communication overhead, and client reliability issues. Additionally, ensuring security and robustness against adversarial attacks remains critical. Despite these challenges, FL offers a promising framework for large-scale privacy-preserving machine learning.

One of the most challenging issues in FL is the presence of noisy clients, particularly those with label noise in their local datasets [4]. Label noise can arise from incorrect annotations, varying expertise levels,

*Corresponding author. Email: hai.tranthithanh1@hust.edu.vn

or inherent ambiguities in data labeling, mostly in the medical field [5], [6], leading to degraded model performance. In FL, this problem is amplified due to the decentralized nature of training, where the server has no direct access to the raw data to verify its quality. Consequently, noisy clients can introduce biased or misleading updates, disrupting the learning process and harming the global model's accuracy. Addressing this issue requires robust methods to detect and mitigate label noise at both the client and server levels, such as noise filtering, re-labeling strategies, or weighting client contributions. Tackling noisy clients is essential for ensuring the reliability and scalability of FL in real-world scenarios.

Existing approaches to deal with noisy clients in FL often focus on two main strategies. The first category aims to identify noisy clients and assign them less importance during the aggregation process [7–9]. Reducing these clients's influence helps minimize noise and improve model performance. The second category attempts to correct the noisy labels provided by these clients [10], [11], [12]. This dual strategy not only mitigates the negative effects of noisy data, but also enhances the robustness of the learning process. By combining client identification with label correction, federated learning systems can maintain higher accuracy and stability, even in the presence of substantial noise. The second category generally requires an additional stage of processing as well as a clean benchmark dataset on the server for validation, which may not be available in practical use. For instance, techniques like co-teaching or confident learning rely on server-side validation datasets to identify mislabeled samples

In this paper, we propose a framework, namely FedNDA (**Fed**erated **N**oisy Client **D**etection and Robust **A**ggregation), which belongs to the first category of federated noise label learning. It aims to detect abnormal clients based on their different loss distributions. To address class imbalance in the data, per-class losses are utilized. Noisy clients with significant variations in their per-class losses may be candidates for having a data distribution that differs from clean clients. FedNDA consists of two main stages: noisy client detection and robust aggregation with training. Detection is based on analyzing the distribution of clients in the per-class loss space. We then assign a weight indicating the importance of clients based on their noisiness. This noisiness is computed as the Earth Mover's Distance (EMD) between the current per-class loss and the mean per-class loss of all clean clients.

Unlike FedNoRo [13], which measures the distance based on the minimum distance from a client to the closest clean client, FedNDA uses EMD which provides several advantages. EMD measures the minimum cost of transforming one probability distribution into another, making it highly sensitive to class distribution differences. It can better detect noisy clients whose class distribution deviates significantly from the global or expected distribution. Additionally, EMD inherently considers label relationships when used in classification problems, particularly in scenarios with ordinal labels or correlated classes. This makes it more robust to label noise, as it penalizes misclassifications proportionally to the severity of the error. In summary, our contributions are three-fold:

- We propose a framework, FedNDA, for robust federated learning with noisy labels;
- We introduce a new metric to measure the noisiness of clients based on EMD of per-class losses;
- We validate our proposed methods on two benchmark datasets, CIFAR-10 and ICH, demonstrating the significant improvements of FedNDA compared to other state-of-the-art methods such as FedAvg [1], FedNoRo[13], FedELC[14], FedCorr[10].

The remainder of this paper is structured as follows. In Section II, we provide an overview of the fundamentals of federated learning, the challenges associated with noisy labels, and algorithms used for generating noisy labels in simulations. In Section III, we describe our framework in detail, focusing on its two main stages: noisy client detection and robust aggregation and training. Section IV presents the experimental datasets and results. Finally, we conclude the paper and propose ideas for future work.

2. BACKGROUND AND RELATED WORKS

2.1. Background

Problem definition. Federated Learning is a distributed machine learning paradigm where models are trained collaboratively across multiple decentralized devices or servers holding local data, without transferring the data to a central server. This approach enhances data privacy, reduces communication costs, and allows diverse data distributions while leveraging collective knowledge for improved model performance.

In an FL framework, there is a set of clients $\mathcal{S} = \{C_1, C_2, .., C_N\}$ participating in the training process. Each client C_k holds the local data $\mathcal{D}_k$ which may contain samples belonging to M classes. Each local dataset $\mathcal{D}_k = \{(\mathbf{x}_k^i, y_k^i)\}_{i=1}^{N_k}$ has N_k samples. It is noted that in an IID setting, the probabilities of a class s in the datasets of clients C_k and C_l are similar. This condition does not hold true in the non-IID setting. The overall objective of FL is solving the optimization problem for

N clients over their own local datasets, which can be formulated as eq. (1):

$$\min_{\mathbf{w}} f(\mathbf{w}) := \sum_{k\in[1,N]} \frac{N_k}{N} F_k(\mathbf{w}). \tag{1}$$

where $F_k(\mathbf{w})$ is the loss of the prediction for samples in $\mathcal{D}_k$ of the client C_k with model parameters $\mathbf{w}$.

In training process, each client C_k trains a local model $\mathbf{w}_k$ using its local dataset $\mathcal{D}_k$, then sends to the server for weight aggregation. In a conventional federated learning, averge aggregation is widely used as a simple yet efffective method [1]. At the communication round t, the global model $\mathbf{w}_G^t$ is computed as eq. (2):

$$\mathbf{w}_G^t \leftarrow \sum_{C_k\in\mathcal{S}_t} \frac{|\mathcal{D}_k|}{\sum_{C_i\in\mathcal{S}_t}|\mathcal{D}_i|}\mathbf{w}_k^t \tag{2}$$

with $\mathcal{S}_t \subseteq \mathcal{S}$ is a subset of the selected clients in round t according to a fraction γ. When $\gamma = 1$, all clients participate in the training process.

2.2. Noisy labels and label noise generation for simulation

Noisy labels are errors, inconsistencies, or inaccuracies in the labels of a training dataset used in artificial intelligence and machine learning. Noisy labels can be caused by human error, sensor errors, or inaccurate search engines. To simulate noisy labels, the true label y^j of a certain sample $\mathbf{x}^j$ is replaced with the corresponding noisy label $\hat{y^j}$. Zhang et al. [15] demonstrated that deep learning models are susceptible to overfitting. In [16], [17], [18], label noise was characterized as noise completely at random (symmetric or uniform label noise), noise at random (asymmetric, pair-flipping, label-dependent noise, or instance-independent noise), noise not at random (instance-dependent noise or semantic label noise). Both noise at random and noise not at random happen when either the labeler is not reliable or when there is intrinsic variability among labelers.

Simulation of label noise, which is commonly encountered in real-world datasets, is crucial for validating FL models before their deployment in practical applications. Xu et al. introduced instance-independant label noise. Their noise model is defined as a function of two parameters, ρ and η_l [10]. Here, ρ denotes the system-wide noise level (noisy client rate), while η_l represents the lower bound for the noise level of a noisy client. The local noise level for each client is randomly sampled from the uniform distribution $\mathcal{U}(\eta_l, 1)$. Specifically, the noise level associated with a client is defined as eq.(3):

$$\eta = \begin{cases} u \sim \mathcal{U}(\eta_l, 1) & \text{, with probability } \rho \\ 0 & \text{, with probability } 1-\rho. \end{cases} \tag{3}$$

To take into account the heterogeneity of label noises in the client's data, Wu et al. constructed a heterogeneous label noise model [13]. The main idea is to consider a sample having noise label when it has a high probability to be misclassified by a classication model than clean labels. Let us define the noisy client rate ρ as the proportion of noisy clients and assume local noise rate η_k for the client C_k. The algorithm to generate the label noise from clients of Wu et al. [13] starts by training a neural network g_k on each client C_k with its original clean data $\mathcal{D}_k$. Then, the trained neural network g_k is used to produce the classification probabilities of all samples belonging to the client. Given each instance $\mathbf{x}_k^j$ in the k-th noisy client C_k and the corresponding classification probability $p(Y|\mathbf{x}_k^j) \in [0,1]^M$, its misclassification probability is determined by:

$$\tilde{p}(\mathbf{x}_k^j) = 1 - p(Y = y_k^j|\mathbf{x}_k^j) \tag{4}$$

and totally $\eta_k N_k$ samples would be chosen as noisy samples based on the normalized misclassification probability $\tilde{p}(\mathbf{x}_k^j) \in [0,1]^{N_k}$. A hard sample, which has a high misclassification probability, is considered a noisy sample. As samples are processed on each client among ρN clients by different neural networks g_k, the samples are ensured to be heterogeneous.

2.3. Related works on FL with label noise

The challenge of noisy clients in FL has garnered significant attention in recent research, focusing on both detection and aggregation strategies. We divide the existing methods on FL with label noise into two categories: Noisy label learning without correction and Noisy label learning with correction.

Noisy label learning without correction. In [7], Chen et al. introduced FOCUS, a model that assigns each local model a credibility measure reflecting the quality of sample labels provided by the clients. The credibility is computed as the sum of cross-entropies when the local and global models are validated on a clean, small benchmark dataset on the server. A higher credibility value indicates that the client may have noisier data. This credibility measure is then used as a weight in the model aggregation process. Similarly, Yang et al. [19] assumed the availability of a small clean dataset on the server and measured the noise ratio of each client based on this clean validation dataset. These methods focus on identifying or assigning a metric to measure the quality of a client without directly correcting the noisy labels. RHFL [20] addresses the complexities of model heterogeneity among clients while managing noise. It aligns the logits output distributions across heterogeneous models and employs a noise-tolerant loss function during local training. This approach helps mitigate the adverse effects of label noise, allowing for more reliable aggregation of client updates.

One notable approach is the FedNoRo method [13], which employs a two-component Gaussian Mixture Model to identify noisy clients based on per-class loss vectors produced by each client. They then define a distance from the noisy client to the closest clean model in the per-class loss space as weight for global model aggregation. Another prominent framework is FedNed framework [9]. It introduces a technique called negative distillation, which encourages the global model's predictions to diverge from those of identified noisy clients. By identifying extremely noisy clients in each communication round, FedNed excludes their contributions from model updates, leading to improved performance. This method incorporates a local optimization strategy for these clients, enhancing overall model accuracy significantly compared to traditional methods. FedNed also utilized a clean benchmark dataset on server to evaluate the noiseness of clients.

Noisy label learning with correction. In the context of machine learning with noisy labels, label noise correction aims to improve the quality of training data by actively modifying potentially incorrect labels. Instead of treating noisy labels as fixed, label correction techniques attempt to replace them with more accurate ones, often leveraging the model's own predictions or other information sources.

Xu et al. proposed FedCorr [10], which uses the Local Intrinsic Dimensionality (LID) score to evaluate client quality during a pre-processing stage. Noisy clients and labels are identified and corrected in a subsequent correction stage, followed by standard FL training in the final stage. Tsouvalas et al. proposed FedLN to deal with noise label [8]. FedLN estimates each client's noise level in a single federated round and enhances model performance by correcting noisy samples or reducing their impact. Li et al. proposed FedDiv an one-stage framework for federated learning with noisy labels [11]. FedDiv leverages complementary knowledge from all clients to train a global noise filter while simultaneously conducting label noise filtering locally on each client. By leveraging knowledge across clients, FedDiv effectively filters label noise and improves training stability. Jiang et al. proposed FedELC framework [14], an end-to-end label correction mechanism that detects high-noisy clients and corrects their labels via backpropagation, improving data quality and model performance. Giap et al. [12] introduced FedDC, a three-stage framework for noisy detection, correction, and standard training, accommodating scenarios where the number of clients participating in the training process is dynamic.

3. PROPOSED METHOD

In this paper, we propose a method, namely FedNDA to deal with noise label without label correction due to its simple design and lower computational complexity. The idea is to identify the noise clients which may negatively impact the performance of the overall system, then reduce its importance in the aggregation step at the server side. FedNDA consists of two main stages:

- **Stage 1: Noisy client detection** aims at identifying noisy clients and also identifying clean clients based on analyzing the distribution of the per-class losses by each client model on its local dataset.
- **Stage 2: Robust aggregation and training** quantify the noisiness of each client using the EMD between its per-class loss vector and the mean per-class loss vector of all clean clients. The server-side aggregation process is then guided by this noisiness score.

Figure 1 illustrates the two stages of FedNDA. We will describe in detail each stage in the following subsections.

3.1. Stage 1: Noisy Client Detection

Step 1: In our framework, all N clients participate in the training process. Firstly, FedNDA begins with a T_1-round warm-up training phase using the FedAvg algorithm. During this phase, each client C_k trains a local model $\mathbf{w}_k$ using its own data $\mathcal{D}_k$ and sends the updated model parameters back to the server for average aggregation. It is noted that $\mathcal{D}_k$ may contain noisy samples. The global model $\mathbf{w}_G$ is then sent back to clients for the next round of training. It is computed as eq. (5):

$$\mathbf{w}_G = \sum_{k=1}^{N} \frac{N_k}{\sum_{j=1}^{N} N_j} \mathbf{w}_k \quad (5)$$

where $\mathbf{w}_G$ and $\mathbf{w}_k$ denote the weights of the global model and the k-th local model respectively, and N is the number of clients for aggregation.

Step 2: After the warm-up phase, each client C_k calculates the per-class loss vector for its local dataset $\mathbf{L}_k = [l_{k1}, l_{k2}, \ldots, l_{kM}]^T$, where M is the number of classes to be recognized. Here, l_{kj} represents the Cross-Entropy loss for the jth class at client C_k. These per-class loss values reflect the model's performance on each class. The per-class loss values from all clients are sent to the server (Step 2 in Fig. 1).

Step 3: On the server side, all N per-class loss vectors from N clients, $\mathbf{L}_1, \mathbf{L}_2, \ldots, \mathbf{L}_N$, are fed into a Gaussian Mixture Model (GMM). GMMs have been widely used in various studies for unsupervised clustering of data points with similar attributes. In the context of

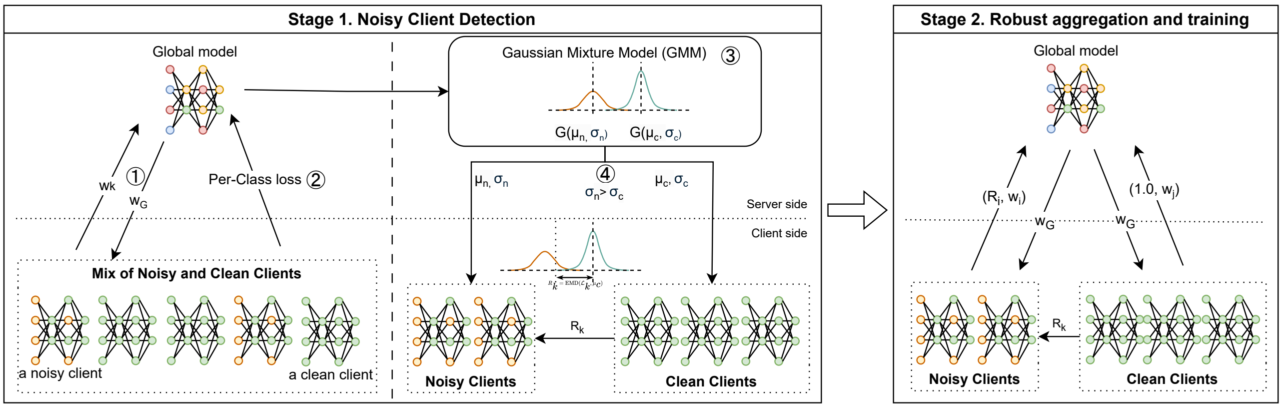

Figure 1. Overview of the FedNDA two stages framework: **Stage 1**: ① Warm-up training with FedAvg though T_1 rounds; ② Computation of per-class losses for all clients and send to the server; ③ Estimation of Gaussian Mixture Model to cluster clean and noisy clients; ④ Estimation of noisiness for each client based on EMD; **Stage 2**: training of noisy and clean clients with EMD based noisiness-aware aggregation.

identifying clean and noisy clients, some prior works have also employed GMMs to separate the two groups, for example, FedNoRo [13], FedRN [21], and FedELC [14]. Inspired by this idea, we also follow this approach to classify clients into two categories.

The GMM aims to analyze clients by grouping them into two Gaussian distributions $\mathcal{G}(\mu_c, \sigma_c)$ and $\mathcal{G}(\mu_n, \sigma_n)$ where (μ_c, σ_c) and (μ_n, σ_n) are the mean and deviation of clean Gaussian and noise Gaussian respectively. Clients without noisy labels have similar loss values, resulting in a small deviation for their Gaussian distribution. In contrast, the loss values of noisy clients are different, leading to a larger deviation in their Gaussian distribution ($\sigma_c < \sigma_n$). In this way, the clients can be clustered into two groups: clean and noisy.

Step 4: Each noisy client C_k, once is identified as noisy will be assigned a *noisiness score* R_k. This value correlates with the noise rate of the client. The higher the noisiness score R_k, the more samples have label noise, resulting in a higher noisy sample rate η_k. In our work, noisiness of a client C_k is defined the EMD between its per-class loss distribution $\mathbf{L}_k$ and the average per-class loss distribution of the clean clients μ_c (eq. (6)).

$$R_k = \text{EMD}(\mathbf{L}_k, \mu_c) = \min_{\mathbf{F}} \sum_{i=1}^{M} \sum_{j=1}^{M} f_{ij} d_{ij} \tag{6}$$

$$\text{subject to} \sum_{j=1}^{M} f_{ij} = \mathbf{L}_{ki}, \quad \sum_{i=1}^{M} f_{ij} = \mu_{cj}, \quad f_{ij} \geq 0 \tag{7}$$

where $\mathbf{F} = \{f_{ij}\}$ is the flow matrix, determined by the optimization function (eq.(6)) and d_{ij} is the ground distance between ith and jth elements.

The μ_c is estimated as the centroid of the clean cluster as follows :

$$\mu_c = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{L}_i \tag{8}$$

where N_c is the number of noisy clients.

3.2. Stage 2: Robust aggregation and training

Specific training for groups of clean / noise clients. In the second stage, we process the training clean and noisy clients as follows:

- **Clean clients** C_c: We utilize the Cross-Entropy loss function to train the clean clients.

$$L_c = L_{\text{CE}}(y_p, \hat{y}) \tag{9}$$

where y_p and $\hat{y}$ are the predicted and ground truth labels respectively. It is to note that Logit Adjustment (LA) is applied to the output of the network (y_p) to reduce the effect from class imbalance and heterogenity of data among clients. This is a technique used to modify the output of a classification model, typically a logistic regression model, to correct for imbalanced class distributions or to optimize for a specific performance metric.

- **Noisy clients** C_n: Beside the Cross-Entropy loss combined with LA technique, we adjust the Kullback-Leibler (KL) divergence (eq. (10)) loss into the total loss to train the noisy clients:

$$L_n = \lambda L_{\text{KL}}(y_p, y_G) + (1 - \lambda) L_{\text{CE}}(y_p, \hat{y}) \tag{10}$$

where y_p represents the prediction results of the local model, L_{KL} is the Kullback-Leibler divergence, and λ is a trade-off coefficient. In our experiment, λ is set to 0.8. For noisy client, with a given sample $\mathbf{x}$, the global model $f_G(\mathbf{x})$ produces a targeted probability distribution y_G calculated as

$$y_G = \text{softmax}\left(\frac{f_G(\mathbf{x})}{T}\right) \tag{11}$$

T is the temperature to control and is set as 0.8 by default.

Algorithm 1 FedNDA algorithm

Input: Number of clients N; set of local data $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_N\}$; number of classes M; rate of noisy clients ρ; the communication rounds for the first and second stages T_1 and T_2
Output: global model $\mathbf{w}_G$

1: **Stage 1: Noisy client detetection**
2: Randomly select a fraction ρ of from the total number of clients N.
3: **for** each selected client C_k **do**
4: Perform the training for T_1 rounds with FedAvg (eq. (5)).
5: Calculate the per-class loss values l_{kj} of client C_k for the class $j \in [1, M]$
6: Send $\mathbf{L}_k = [l_{k1}, l_{k2}, \dots, l_{kM}]^T$ to the server
7: **end for**
8: At the server: fed $\{\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_K\}$ to the GMM module to estimate two Gaussian distributions: $\mathcal{G}(\mu_c, \sigma_c)$ and $\mathcal{G}(\mu_n, \sigma_n)$
9: Identify the noisy clients in the noisy distribution $\mathcal{G}(\mu_n, \sigma_n)$ with $\sigma_n > \sigma_c$.
10: Estimate the noisiness value R_k for each client according to eq. (6).
11: **Stage 2: Robust aggregation and training**
12: **for** round $t \in [1, T_2]$ **do**
13: Train clean clients with Cross-Entropy loss function and Logit Adjustment eq. (8).
14: Train noisy clients with Cross-Entropy loss function, Logit Adjustment and Kullback-Leibler divergence (eq. (9)).
15: Update the global model $\mathbf{w}_G^t$ with noisiness aware aggregation (eq. (12))
16: **end for**

Robust aggregation. To further reduce the negative impact of noisy clients, a distance-aware model aggregation function is used. Differ from other approaches such as FedNoRo [13] that weights clients based on the distance of their models from the nearest clean client's model, in our FedNDA, noisiness is integrated into an aggregation process. As mentioned previously, EMD quantifies the minimum cost required to transform one probability distribution into another, making it highly sensitive to differences in class distributions. This sensitivity allows it to more effectively identify noisy clients whose class distributions differ significantly from the global or expected distribution. This is done by combining the EMD-based weight for each client, ensuring that both data quality (as reflected by EMD) and model convergence influence the client's contribution to the global model. To ensure the boundness of R_k, it is further normalized to $[0, 1]$ as

$$\overline{R_k} = \frac{R_k}{\max_j R_j} \tag{12}$$

Then, local models are aggregated to update the global model by

$$\mathbf{w}_G = \sum_{k=1}^{N} \frac{\delta_i}{\sum_{j=1}^{N} \delta_j} \cdot \mathbf{w}_k \tag{13}$$

where

$$\delta_k = \begin{cases} 1.0, & \text{if } C_k \text{ is a clean client} \\ e^{-\overline{R_k}} & \text{if } C_k \text{ is a noisy client} \end{cases} \tag{14}$$

The steps of the first and the second stage are summarized in Algorithm 1. Totally, we train FedNDA for $T_1 + T_2$ rounds, T_1 rounds for the first stage and T_2 rounds for second stage of the framework.

4. Experiments

4.1. Datasets

Overview of the datasets. To evaluate the performance of FedNDA compared to other existing methods such as FedAvg and FedNoRo, we utilize two benchmarks: CIFAR-10 [22] and ICH [23]. In CIFAR-10, the main task is the image classification of ten classes. The CIFAR-10 dataset consists of 60,000 color images of size 32x32 across 10 classes. Each class contains 6,000 images, representing categories such as airplanes, cars, birds, cats, deer, dogs, frogs, horses, rabbits, and trucks. 50,000 samples are used for training and 10,000 are used for testing. While CIFAR-10 is frequently used to assess federated learning frameworks, the ICH dataset has been specifically designed as a comprehensive resource for evaluating Brain CT Hemorrhage classification methods. The ICH dataset comprises 67,969 brain CT slices and includes five classes: subarachnoid, intraventricular, subdural, epidural, and intraparenchymal hemorrhages, which are commonly observed in brain CT scans. In the ICH dataset, samples are randomly divided into the training and test sets following a 7:3 split. Figure 2 and Figure 3 illustrate samples of the two datasets respectively.

Data partition across clients. In this experiment, we focus on evaluation with non-idd setting. To allocate data to clients, we utilize both Bernoulli and Dirichlet distributions. The Bernoulli distribution controls the probability of retaining data samples for each client, while the

Figure 2. Samples from CIFAR-10 dataset.

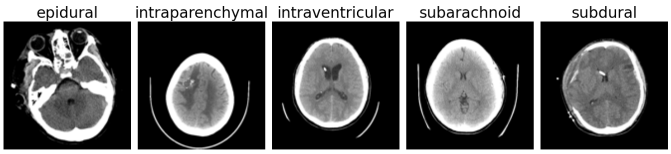

Figure 3. Samples from ICH dataset.

Dirichlet distribution allocates data classes unevenly across clients. This combination effectively simulates real-world federated learning scenarios where clients have heterogeneous data in both sample size and class distribution. As a result, it provides a robust environment for evaluating the generalization capabilities and efficiency of federated learning algorithms.

For this partition, we generate the indicator matrix Φ of size $N \times M$, where N is the number of clients, and M is the number of classes in the dataset. The element Φ_{ij} indicates whether the local dataset $\mathcal{D}_i$ of the client C_i contains the class $j \in [1, M]$. The value of Φ_{ij} is determined by Bernoullie distribution with probability p. We define the vector $\mathbf{z}_j$ with length of $\sum_{i=1}^{N} \Phi_{ij}$ that is the total number of clients containing the class j. The element of the vector $\mathbf{z}_j$ will be sampled from symmetric Dirichlet distribution α_{dir}. In our experiments, N is set to 20, and M is 10 and 5 regarding CIFAR-10 and ICH dataset, respectively. We choose $p = 0.9$ for Bernoulli distribution and $\alpha_{dir} = 2.0$ for Dirichlet distribution.

Noisy label generation. We begin by defining the Noisy client rate ρ, which represents the proportion of clients with noisy samples among the total N clients. Specifically, the number of clients with noisy samples is ρN. Additionally, we define the local noise rate η_k for each client C_k following a uniform distribution $\mathcal{U}(\eta^l, \eta^u)$, where η^l, η^u are lower and upper noisy sample rates, respectively. In our experiments, we set ρ to 0, 0.2, 0.4, 0.6, 0.8, and 1, representing noise levels ranging from no noise to very high levels of noisy clients, and (η^l, η^u) with (0.3, 0.5) and (0.5, 0.7). Based on these rates, we randomly generate noisy labels according samples of clients.

Figure 4 and Figure 5 illustrate the data distributions for $N = 20$ clients for the CIFAR-10 and ICH datasets, respectively. Colors represent the classes distributed across each client, while the length of each bar represents the number of samples for each class. We clearly observe the non-IID nature of the data thanks to the use of data sampling based on Bernoulli and Dirichlet distributions.

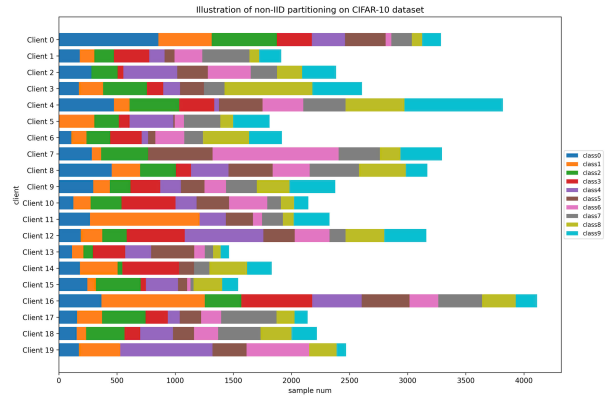

Figure 4. Illustration of non-IID partitioning on CIFAR-10 dataset.

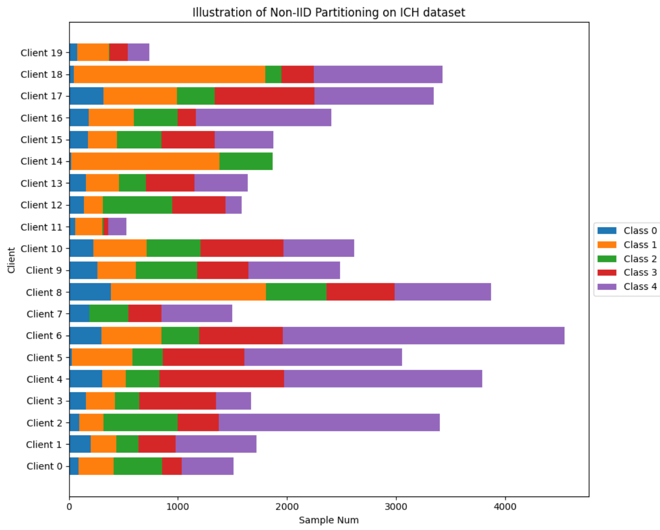

Figure 5. Illustration of non-IID partitioning on ICH dataset.

4.2. Implementation details

We implement ResNet-18 [24] with a pre-trained initialization from ImageNet. The number of communication rounds is set to 100, and the local epoch is 5. The global model warm-up phase, denoted as T_1, is set to 15 rounds using FedAvg before initiating noisy client detection. We set a constant learning rate l_r of 3e-4, and a batch size b of 16. Table 1 summarizes

Table 1. List of hyper-parameters and models used in our experiments.

Parameters	Symbol	Value
Number of clients	N	20
Number of classes (CIFAR-10/ICH)	M	10/5
Model architecture	F	ResNet-18
Participing client rate	γ	1
Noisy client rate	ρ	{0, 0.2,..., 1}
Upper noisy sample rate	η^u	0.5, 0.7
Lower noisy sample rate	η^l	0.3, 0.5
Noisy sample rate	η_k	$Uniform(\eta^l, \eta^u)$
Bernoulli distribution's prob.	p	0.9
Symmetric Dirichlet distribution's prob.	α_{dir}	2.0
1st stage communication rounds	T_1	15 for CIFAR-10 10 for ICH
2nd stage communication rounds	T_2	85 for CIFAR-10 90 for ICH
Learning rate	l_r	3e-4
Weight decay	w_d	5e-4
Trade-off coefficient	λ	0.8
Temperature value	T	0.8
Batchsize	b	16

the most important hyperparameters utilized in our experiments.

4.3. Experimental results

The experiments have been conducted to evaluate: 1) the ability of FedNDA for noisy client detection; 2) the overall performance of FedNDA compared to two state-of-the-art models: FedAvg (without noisy client detection) and FedNoRo (with noisy client detection); 3) the robustness of FedNDA to the noisy level.

Evaluation of noisy client detection. Evaluating the accuracy of noisy client detection is crucial, as it significantly impacts subsequent processes. In our noisy label generation process, we have ground truth information about which clients are noisy and their respective noise levels. By using GMM as a classifier for noisy and non-noisy clients, we can identify the noisy cluster and the clean cluster based on the prediction results.

We first represent each client as a feature point in the space defined by per-class loss vectors. Principal Component Analysis (PCA) is applied, retaining the two most significant components for visualization, as shown in Figure 6. This experiment is conducted on CIFAR-10 as an example. Clearly, clean clients (blue circles) exhibit similar per-class loss patterns, clustering closely in the feature space. In contrast, noisy clients (red circles) show diverse per-class loss patterns and are distributed sparsely throughout the feature space. This demonstrates that using a Gaussian Mixture Model, combined with per-class loss vectors, is highly effective in distinguishing clean clients from noisy ones.

We further investigate the ability to estimate the noisiness of clients. In this experiment, we evaluate the effectiveness of the proposed Earth Mover's Distance (EMD) approach with the conventional Euclidean distance (ED) for measuring noisiness score. Figure 7 presents the noisiness normalized to the range $[\eta^l, \eta^u]$ alongside the ground truth noise level η^k for each noisy client C_k, previously identified using GMM. The results indicate that the normalized noisiness closely aligns with the true noisy sample rate of each client. Notably, for client C_{17}, both ED and EMD perfectly estimate the noisiness. However, EMD appears to provide slightly more accurate estimations overall.

Figure 8 shows the performance of noisy client detection at different noisy client rates $\rho = 0.2, 0.4, 0.6, 0.8, 1$. It is interesting to see that our FedNDA is able to detect with 100% of accuracy when $\rho = 0.2, 0.4, 0.6, 0.8$. When all clients are noisy $\rho = 1$, the accuracy decreases to 71.43%. However, this situation does not always occur in practice.

To evaluate the performance of the GMM-based method for identifying clean and noisy clients, we conducted K-Means (K = 2) and DBSCAN as two additional clustering methods in step 3 of the first stage of our algorithm. Using DBSCAN, we set the neighborhood radius (ϵ) to 0.5 and the minimum number of points (MinPts) to 2. Fig. 9 shows that GMM and K-Means produce similar results of detection, leading to the same accuracy graphs, which overlap with each other. DBSCAN starts slower than GMM and K-Means, but finally reaches comparable accuracy.

Comparison with existing models. In this paper, we compare the performance of our proposed model, FedNDA, with existing models. On the CIFAR-10 dataset, we re-implement and train the experimented models (FedAvg, FedCorr, FedELC, FedNoro) while on the ICH dataset, we utilize the reported results from existing works (FedAvg, FedProx, FedLA, RoFL, RHFL, FedLSR, FedCorr, FedNoRo) with the same experimental setup.

Results on the CIFAR-10 dataset Experiments on this dataset were performed under two different configuration settings: 1) the first setting employs the SGD optimizer with image resizing and reports the global accuracy after 100 rounds; and 2) the second setting employs the Adam optimizer without image resizing and reports the result after 50 rounds.

Table 2 shows the comparative result of FedNDA versus FedAvg[1], FedCorr[10], FedELC[14], and FedNoRo[13] on the CIFAR-10 dataset. The results show that FedNDA consistently achieves the highest

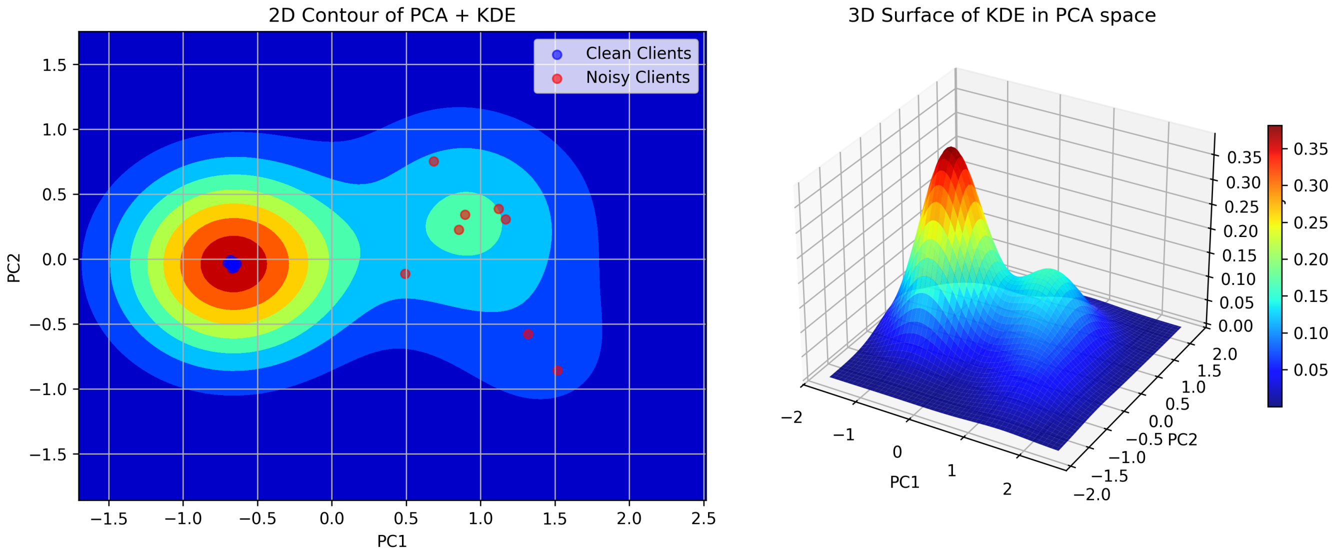

Figure 6. Visualization of GMMs estimated with 20 clients using the first two components of the Principal Component Analysis (PCA) and Kernel Density Estimation (KDE) algorithms on the CIFAR-10 dataset.

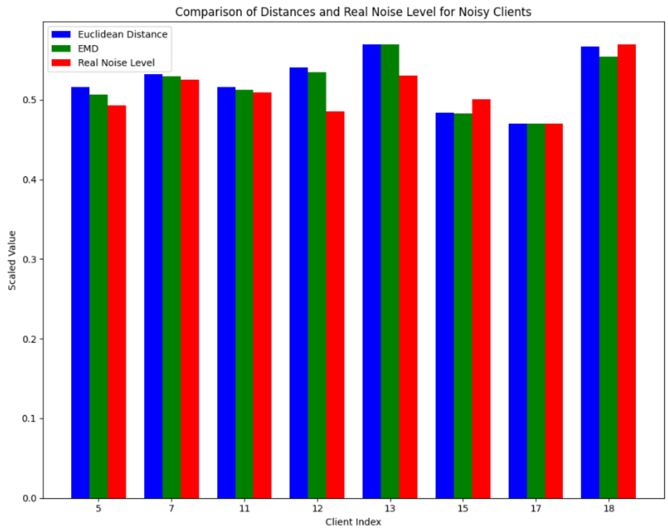

Figure 7. Evaluation of estimation of noisy sample rate at each client

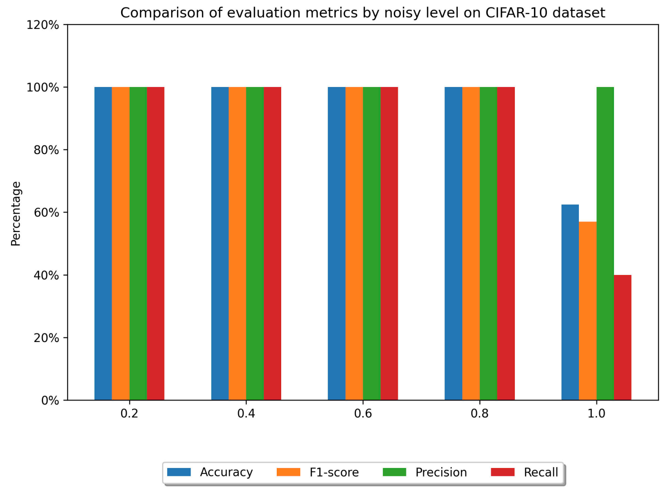

Figure 8. Performance of noisy client detection at different noise noisy client rates ρ.

accuracy across all experimental setups, demonstrating its robustness and adaptability. Specifically, when using the SGD optimizer with resized inputs, FedNDA achieves 84.99% of accuracy without noisy clients and 84.38% with noisy clients, outperforming FedAvg and FedNoRo by notable margins. When using the Adam optimizer without resizing, FedNDA further improves, reaching 87.39% of accuracy without noise and 86.13% with noise, maintaining its leading position even as other methods experience performance drops. FedCorr and FedELC generally lag behind, with FedCorr being particularly sensitive to the presence of noisy clients,

Table 2. Comparison of FedNDA's accuracy (BACC) with existing models in both testing scenarios on CIFAR-10 dataset: without noisy clients $(\rho = 0, (\eta^l, \eta^u) = (0.0, 0.0))$ and with noisy clients $(\rho = 0.4, (\eta^l, \eta^u) = (0.3, 0.5))$.

Method	Without noisy clients		With noisy clients	
	SGD Resize 100 rounds	Adam No Resize 50 rounds	SGD Resize 100 rounds	Adam No Resize 50 rounds
FedAvg [1]	79.26	-	79.70	-
FedCorr [10]	-	75.31	-	69.53
FedELC [14]	-	84.85	-	83.53
FedNoRo [13]	81.95	86.94	80.78	86.05
FedNDA (our)	**84.99**	**87.39**	**84.38**	**86.13**

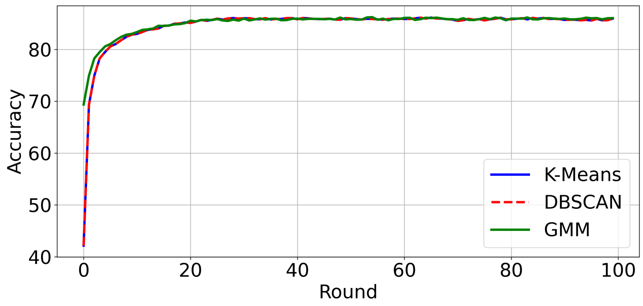

Figure 9. Global accuracy on the validation set with various clustering algorithms for FedNDA on the CIFAR-10 dataset: GMM, DBSCAN, K-Means in an experimental setting $\rho = 0.4, (\eta^l, \eta^u) = (0.3, 0.5)$.

and FedELC showing moderate resilience, especially under Adam optimization.

Overall, the introduction of noisy clients leads to accuracy reductions for all methods, but FedNDA's performance remains the most stable, indicating its strong resistance to data corruption and client heterogeneity. Fig. 10 and Fig. 11 illustrate the FedNDA's accuracy across training rounds on the CIFAR-10 dataset without noisy and with noisy clients, respectively. Experiments show that FedNDA achieves more stable results and converges quickly than FedELC and FedCorr.

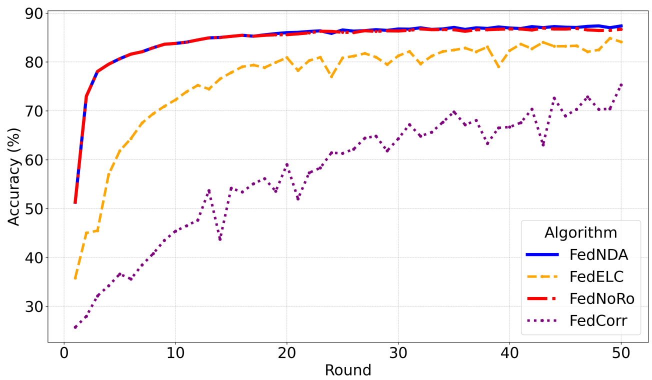

Figure 10. Comparison of algorithm accuracy over rounds without noisy client

Results on the ICH dataset: Table 3 presents the comparative results of FedNDA on the ICH dataset. In addition to comparisons with FedAvg and FedNoRo, we also include results from existing works. For without noisy client setting (i.e $\rho = 0, (\eta^l, \eta^u) = (0.0, 0.0)$), FedAvg achieved 69.34% of accuracy, while the accuracy increased by 4.25% with FedNoRo. Our FedNDA provides the highest

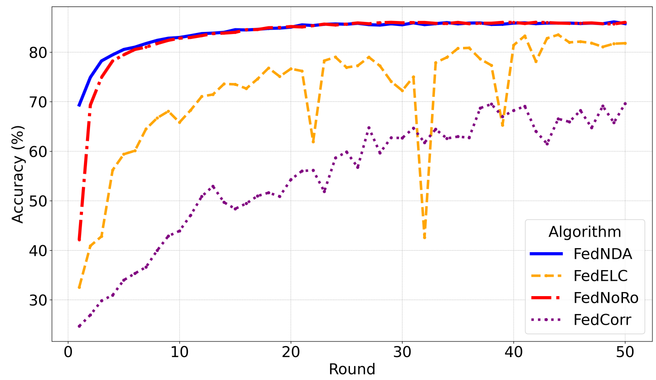

Figure 11. Comparison of algorithm accuracy over rounds with noisy client rate at 0.4

accuracy of 73.81%, which is 4.47% and 0.22% higher than FedAvg and FedNoRo, respectively. With noisy client setting, we report the result with the rate of noisy clients $\rho = 0.4$ and the local noise rate of samples η_i following $\mathcal{U}(\eta^l, \eta^u) = (0.3, 0.5)$, FedNDA stills outperformed FedNoRo and other FL models such as FedAvg, FedProx, FedLA, ROFL, FedLSR, FedCorr.

Table 3. Comparison of FedNDA's accuracy (BACC) with existing models in both testing scenarios on ICH dataset: without noisy clients ($\rho = 0, (\eta^l, \eta^u) = (0.0, 0.0)$) and with noisy clients ($\rho = 0.4, (\eta^l, \eta^u) = (0.3, 0.5)$).

Method, Year	Without noisy clients	With noisy clients
FedAvg [1]	69.34	60.52
FedProx [25]	68.16	60.85
FedLA [26]	73.56	66.60
RoFL [27]	-	40.35
RHFL [28]	-	55.26
FedLSR [29]	-	52.48
FedCorr [10]	-	53.62
FedNoRo [13]	73.59	70.69
FedNDA (our)	**73.81**	**71.17**

Robustness of FedNDA to noisy client rate. To evaluate the robustness of FedNDA to different noise client rates, we vary the rate of noisy clients ρ from 0.2 to 1 with a step size of 0.2. We also change the rate of noisy samples η^k in each client C_k using a uniform distribution $\mathcal{U}(\eta^l, \eta^u)$ with η^l, η^u are lower and upper bounds of the noisy sample rate. In our experiment, $(\eta^l, \eta^u) = (0.3, 0.5)$ and $(\eta^l, \eta^u) = (0.5, 0.7)$. Fig. 12 shows the accuracy of FedNDA for these two different pairs of (η^l, η^u). All experiments are conducted on the CIFAR-10 dataset. We observe that the accuracy of FedNDA remains consistent as the number of noisy clients increases. When all clients are noisy ($\rho = 1$), the accuracy decreases slightly; however, this scenario may not be realistic in practice, as it is unlikely that all clients are noisy.

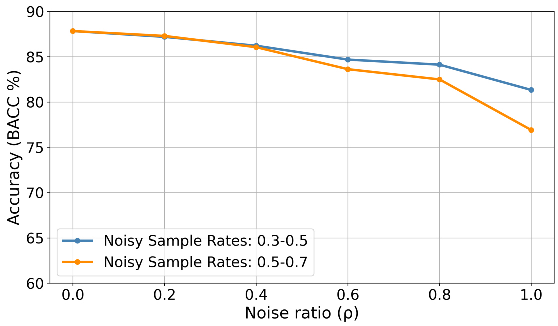

Figure 12. Robustness of FedNDA to noise client rate ρ when $(\eta^l, \eta^u) = (0.3, 0.5)|(0.5, 0.7)$ on CIFAR-10 dataset. The rate of noisy clients ρ (horizontal axis) varies from 0.2 to 1. Values in the vertical axes represent the corresponding accuracies (%).

Impact of λ on the FedNDA performance. As mentioned in section 3.2, we applied a specific training strategy for each group of clients. For clean clients, we utilized the Cross-Entropy loss function, while we combined CE loss with KL loss to train noisy clients. In eq.(10), the hyperparameter λ defines the weight for each loss component. We varied the value of λ from 0.2 to 1 with a step size of 0.2. A higher value of λ increases the contribution of the KL loss to the overall loss function. Fig. 13 shows the accuracy achieved by FedNDA across different values of λ. We observe that $\lambda = 0.2$ yields the highest accuracy of 86.5%. When $\lambda = 1$, meaning KL is not considered, the accuracy is reduced to 86.05%. This result is reasonable, as it reflects a balanced contribution of the KL loss to the overall objective function.

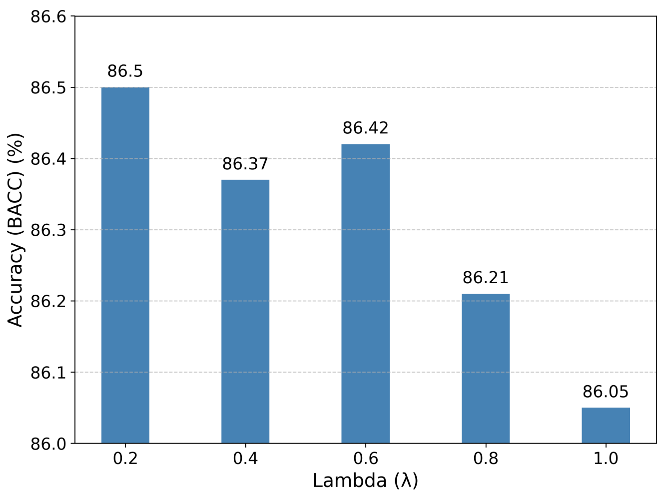

Figure 13. The effect of on model robustness under random label noise on the CIFAR10 dataset

5. Conclusions

This paper introduced a novel framework, FedNDA, for federated learning. The proposed two-stage framework can identify noisy clients based on their per-class loss vectors using the Gaussian Mixture Model (GMM) technique. It then determines the noise rate of identified noisy clients by computing the Earth Mover's Distance (EMD) between the distribution of per-class losses and that of the average losses from clean clients. In the first stage, our method achieves 100% of accuracy in determining the noise rate when noise levels range from 0.2 to 0.8. The accuracy decreases to 71.43% when all clients are noisy. In the second stage, the framework trains clean and noisy clients differently, employing a noise-aware aggregation strategy. Our approach outperforms state-of-the-art FL algorithms on two benchmark datasets. Additionally, the method demonstrates high stability across varying noisy client rates. In future work, we plan to integrate more information, such as gradient information or Local Intrinsic Dimensions (LID), to better identify noisy clients when all clients are noisy. We also aim to evaluate the framework on real-world noisy datasets.

Acknowledgements

This research is funded by the Ministry of Education and Training (MOET) under grant number B2023-BKA-09 "Research and development of supporting tools for prognosis of traumatic brain injury using multi-modal information and artificial intelligence"

References

[1] McMahan HB, Moore E, Ramage D, y Arcas BA. Federated learning of deep networks using model averaging. arXiv preprint arXiv:160205629. 2016;2(2).

[2] Li T, Sahu AK, Talwalkar A, Smith V. Federated learning: Challenges, methods, and future directions. IEEE signal processing magazine. 2020;37(3):50-60.

[3] Li Q, Wen Z, Wu Z, Hu S, Wang N, Li Y, et al. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. IEEE Transactions on Knowledge and Data Engineering. 2021;35(4):3347-66.

[4] Song H, Kim M, Park D, Shin Y, Lee JG. Learning from noisy labels with deep neural networks: A survey. IEEE transactions on neural networks and learning systems. 2022;34(11):8135-53.

[5] Irvin J, Rajpurkar P, Ko M, Yu Y, Ciurea-Ilcus S, Chute C, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33; 2019. p. 590-7.

[6] Karimi D, Dou H, Warfield SK, Gholipour A. Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. Medical image analysis. 2020;65:101759.

[7] Yiqiang C, Xiaodong Y, Xin Q, Han Y, Biao C, Zhiqi S. FOCUS: Dealing with label quality disparity in federated learning. arXiv preprint arXiv:200111359. 2020.

[8] Tsouvalas V, Saeed A, Ozcelebi T, Meratnia N. Labeling Chaos to Learning Harmony: Federated Learning with Noisy Labels; 2023. Available from: https://arxiv.org/abs/2208.09378.

[9] Lu Y, Chen L, Zhang Y, Zhang Y, Han B, ming Cheung Y, et al.. Federated Learning with Extremely Noisy Clients via Negative Distillation; 2024. Available from: https://arxiv.org/abs/2312.12703.

[10] Xu J, Chen Z, Quek TQS, Chong KFE. FedCorr: Multi-Stage Federated Learning for Label Noise Correction; 2022. Available from: https://arxiv.org/abs/2204.04677.

[11] Li J, Li G, Cheng H, Liao Z, Yu Y. FedDiv: Collaborative Noise Filtering for Federated Learning with Noisy Labels; 2024. Available from: https://arxiv.org/abs/2312.12263.

[12] Giap TT, Kieu TD, Le TL, Tran TH. FedDC: Label Noise Correction With Dynamic Clients for Federated Learning. IEEE Internet of Things Journal. 2024.

[13] Wu N, Yu L, Jiang X, Cheng KT, Yan Z. FedNoRo: Towards Noise-Robust Federated Learning by Addressing Class Imbalance and Label Noise Heterogeneity; 2023. Available from: https://arxiv.org/abs/2305.05230.

[14] Jiang X, Sun S, Li J, Xue J, Li R, Wu Z, et al. Tackling Noisy Clients in Federated Learning with End-to-end Label Correction. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. CIKM '24. ACM; 2024. p. 1015–1026. Available from: http://dx.doi.org/10.1145/3627673.3679550.

[15] Zhang C, Bengio S, Hardt M, Recht B, Vinyals O. Understanding deep learning (still) requires rethinking generalization. Commun ACM. 2021 Feb;64(3):107–115. Available from: https://doi.org/10.1145/3446776.

[16] Frenay B, Verleysen M. Classification in the Presence of Label Noise: A Survey. IEEE Transactions on Neural Networks and Learning Systems. 2014;25(5):845-69.

[17] Han B, Yao Q, Liu T, Niu G, Tsang IW, Kwok JT, et al.. A Survey of Label-noise Representation Learning: Past, Present and Future; 2021. Available from: https://arxiv.org/abs/2011.04406.

[18] Song H, Kim M, Park D, Shin Y, Lee JG. Learning From Noisy Labels With Deep Neural Networks: A Survey. IEEE Transactions on Neural Networks and Learning Systems. 2023;34(11):8135-53.

[19] Yang M, Qian H, Wang X, Zhou Y, Zhu H. Client selection for federated learning with label noise. IEEE Transactions on Vehicular Technology. 2021;71(2):2193-7.

[20] Fang X, Ye M. Robust Federated Learning with Noisy and Heterogeneous Clients. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 10062-71.

[21] Kim S, Shin W, Jang S, Song H, Yun SY. FedRN: Exploiting k-reliable neighbors towards robust federated learning. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management; 2022. p. 972-81.

[22] Krizhevsky A. Learning Multiple Layers of Features from Tiny Images. Master's thesis, University of Tront. 2009.

[23] Flanders AE, Prevedello LM, Shih G, Halabi SS, Kalpathy-Cramer J, Ball R, et al. Construction of a machine learning dataset through collaboration: the RSNA 2019 brain CT hemorrhage challenge. Radiology: Artificial Intelligence. 2020;2(3):e190211.

[24] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition; 2015. Available from: https://arxiv.org/abs/1512.03385.

[25] Li Q, He B, Song D. Model-contrastive federated learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2021. p. 10713-22.

[26] Menon AK, Jayasumana S, Rawat AS, Jain H, Veit A, Kumar S. Long-tail learning via logit adjustment. arXiv preprint arXiv:200707314. 2020.

[27] Yang S, Park H, Byun J, Kim C. Robust federated learning with noisy labels. IEEE Intelligent Systems. 2022;37(2):35-43.

[28] Fang X, Ye M. Robust federated learning with noisy and heterogeneous clients. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022. p. 10072-81.

[29] Jiang X, Sun S, Wang Y, Liu M. Towards federated learning against noisy labels via local self-regularization. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management; 2022. p. 862-73.