

Collaboratively Learning Federated Models from Noisy Decentralized Data

Citation for published version (APA):

Li, H., Funk, M., Gürel, N. M., & Saeed, A. (2025). Collaboratively Learning Federated Models from Noisy Decentralized Data. In W. Ding, C.-T. Lu, F. Wang, L. Di, K. Wu, J. Huan, R. Nambiar, J. Li, F. Ilievski, R. Baeza-Yates, & X. Hu (Eds.), *Proceedings - 2024 IEEE International Conference on Big Data, BigData 2024* (pp. 7879-7888). Article 10825502 Institute of Electrical and Electronics Engineers.
<https://doi.org/10.1109/BigData62323.2024.10825502>

Document license:

TAVERNE

DOI:

[10.1109/BigData62323.2024.10825502](https://doi.org/10.1109/BigData62323.2024.10825502)

Document status and date:

Published: 16/01/2025

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Collaboratively Learning Federated Models from Noisy Decentralized Data

Haoyuan Li¹, Mathias Funk¹, Nezihe Merve Gürel², Aaqib Saeed¹

¹Eindhoven University of Technology, Eindhoven, Netherlands

²Delft University of Technology, Delft, Netherlands

{h.y.li, m.funk, a.saeed}@tue.nl, n.m.gurel@tudelft.nl

Abstract—Federated learning (FL) has emerged as a prominent method for collaboratively training machine learning models using local data from edge devices, all while keeping data decentralized. However, accounting for the quality of data contributed by local clients remains a critical challenge in FL, as local data are often susceptible to corruption by various forms of noise and perturbations, which compromise the aggregation process and lead to a subpar global model. In this work, we focus on addressing the problem of noisy data in the input space, an under-explored area compared to the label noise. We propose a comprehensive assessment of client input in the gradient space, inspired by the distinct disparity observed between the density of gradient norm distributions of models trained on noisy and clean input data. Based on this observation, we introduce a straightforward yet effective approach to identify clients with low-quality data at the initial stage of FL. Furthermore, we propose a noise-aware FL aggregation method, namely Federated Noise-Sifting (FedNS), which can be used as a plug-in approach in conjunction with widely used FL strategies. Our extensive evaluation on diverse benchmark datasets under different federated settings demonstrates the efficacy of FedNS. Our method effortlessly integrates with existing FL strategies, enhancing the global model's performance by up to 13.68% in IID and 15.85% in non-IID settings when learning from noisy decentralized data.

Index Terms—federated learning, decentralized AI, data quality, data-centric machine learning

I. INTRODUCTION

Recent advances in machine learning (ML) have led to a surge in data generated by edge devices. Nonetheless, the ubiquity and heterogeneity of data across these devices pose a significant challenge to the efficacy of training generalizable models. In response to this pressing need, federated learning (FL) is gaining traction as a decentralized paradigm in the field of distributed ML. In FL, the model is learned in a decentralized manner where each client device collaboratively trains a global model without directly sharing the local data with the centralized server [1], [2].

Despite this, one of the primary challenges in ML is data quality, which directly impacts the performance and reliability of ML models. The complexity and nature of data in the context of deep models, which require large-scale data, significantly amplify these challenges [3]. The multifaceted nature of data quality can be compromised in both the input and feature space, leading to issues such as data incompleteness, feature corruption, and label inconsistency [4]. Notably, the issue of maintaining high-quality data becomes particularly

challenging in FL due to its decentralized nature. FL allows to mitigate domain-specific concerns such as IP security, by imposing the requirement that the server has no visibility into client data, thereby increasing the difficulty of ensuring data quality. Moreover, FL is generally vulnerable to (often non-malicious) client failures, especially when it comes to the noisy model update, which can be seen as a special case of data perturbation [5]. These failures typically occur on the client side, where the server has no control over. Specifically, unreliable clients unintentionally distort the generalization of the global model by learning from noisy samples or labels with no control over the data collection phase (e.g., due to no direct interest by the federated client). In practice, the utilization of edge devices for FL is a double-edged sword where a large amount of data can be harvested to learn the model, but such data also often entails significant noise contamination. Particularly for tasks of interest like object detection, data collected from image sensors are susceptible to visual distortions, often attributed to the client's lack of technical expertise or environmental interference [6].

In this work, we mainly focus on the problem of noisy data in the input space, where the features in the client data are (non-maliciously) corrupted. To address this, we first investigate the gradient of the loss function, aiming to detect the presence of noisy client data. Our motivation stems from the fact that the gradient space of the model provides an informative signal regarding the usefulness of the data [7] as it directly captures joint information between representations and output [8]. Inspired by this finding, we pose the question *does the gradient provide a meaningful way to distinguish between models of noisy and clean federated clients?* From the density of the gradient norm, we see a distinct disparity between models trained on noisy and clean input data. We thus exploit the richness of gradient space to propose a straightforward yet effective approach to identify clients with low-quality data at the beginning stage of FL. We further propose a noise-aware FL aggregation method, namely *Federated Noise Sifting* (FedNS). Specifically, it re-weights the contributions of client models based on the data quality (i.e., clean or noisy input) while aggregating a global model (see Figure 1). FedNS can be easily used as a plug-in approach in conjunction with widely used FL strategies. To demonstrate the efficacy of FedNS, we perform extensive experiments on diverse benchmark datasets under different heterogeneous settings. Our findings show that

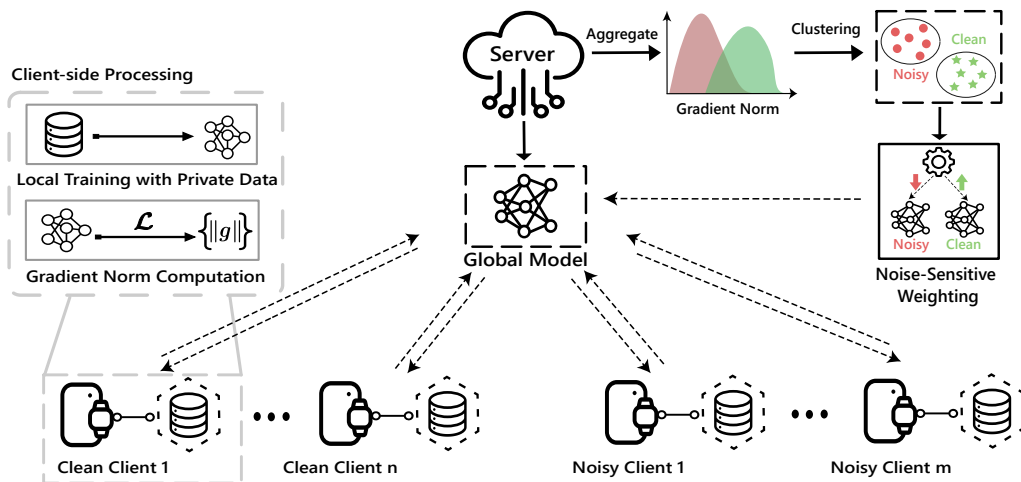


Fig. 1: Overview of FedNS. We propose a plug-in noise-sensitive gradient-norm weighted federated aggregation approach for training high-quality deep models from decentralized data.

our proposed method effortlessly integrates and works well with existing FL aggregation strategies, such as FedAvg [2], FedProx [9], FedTrimmedAvg [10], and FedNova [11] which makes it widely applicable.

Main contributions: Our work makes the following key contributions:

- **Novel single-interaction approach for identifying noisy clients.** We propose an approach that employs gradient norm analysis to identify and categorize noisy (low-quality model updates) clients at the initial training stage (i.e., a first-round) in a federated context without revealing individual client data for inspection.
- **Robust and plug-in weights aggregation strategy.** We introduce a robust and plug-in weights aggregation strategy to mitigate the adverse impacts of noisy clients, named FedNS. It offers a versatile and easily integrable solution to enhance the global model’s generalization. We further substantiate the effectiveness of FedNS through its application to corrupted client data across six datasets and four FL strategies.
- **Systematic conceptualization of noisy FL and benchmark.** We present a concept of noisy FL to understand non-malicious client-side data corruption. Based on this concept, we build¹ the noisy datasets benchmark specifically tailored for FL environments with noisy input.

II. RELATED WORK

Dealing with data quality issues is a well-studied problem in machine learning (ML). Significant efforts have been devoted to mitigating the adverse effect of low-quality data on the generalization capabilities of ML models. [12], [6], and [13] reveal that while Deep Neural Networks (DNNs) exhibit advanced generalization abilities but are notably susceptible to low-quality data, particularly to distortions like noise and blur considerably diminishing their performance. To tackle this challenge, previous work enhances DNN robustness to

image distortions by selectively correcting the activations of the most noise-susceptible convolutional filters [14]. Likewise, [15] treat image distortion identification as multi-label learning and train a multi-task DNN model to improve the model generalization. Instead of reformulating the architecture of the DNN to improve the robustness of the model, [16] proposes to stabilize the training progress against the input perturbations. In a complementary study, [17] shows that fine-tuning and re-training the model using noisy data can effectively alleviate the effect of image distortion. In addition to this, numerous studies have concentrated on addressing label noise, a form of low-quality data in the label space. Training DNNs on noisy labels tends to overfit the noise patterns, which causes degradation in generalization performance on unseen data. Most of the existing methods solve the representation learning with noisy labels by estimating the noise transition matrix and innovatively combining methods like adaptation layers, loss corrections, and regularization techniques [18]–[21]. While these approaches yield substantial results in centralized learning, their applicability is notably limited in federated learning (FL) scenarios, where data is distributed across myriad devices, and the data quality from each client can not be guaranteed due to its private nature.

Recent works in FL mainly focus on addressing the data quality issues pertaining to the label space. Many strategies have been developed to deal with the disparity of label quality in FL. These methods mitigate the impact of label noise by conducting either client selection to re-weight model updates [22]–[24] or data sampling for label correction or exclusion [25]–[29]. Additionally, other works tackle this challenge by correcting the label error. [28] perform label noise correction using consensus-derived class-wise information for dynamic noise identification and label correction. [26] tackles label noise in federated learning by updating local models with globally aligned centroids and correcting labels through global model insights. In spite of the progress that has been made in resolving label noise, data subset selection [7], data valuation [30], [31], dealing with low-quality data in input space

¹we will release the code upon acceptance.

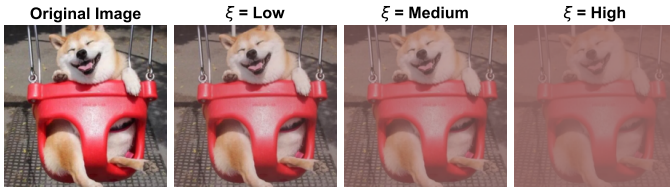


Fig. 2: Impact of *contrast* noise on an image with different levels of noise severity.

(i.e., when noise is in the samples like in images) in the federated setting still remains unexplored. Furthermore, we recognize that client data may be susceptible to backdoor attacks via adversarial methods during both inference and training time [32], [33]; these concerns fall outside the scope of our investigation. To this end, our method aims for a flexible and efficient solution suitable for a range of FL strategies to learn robust models using decentralized data.

III. METHOD

We introduce our robust federated learning (FL) method, named *Federated Noise Sifting* (FedNS), designed for learning from noisy input. We begin with defining the challenge of FL in the context of input noise and then describe our single-interaction approach to discover and classify noisy and clean clients. Then, we propose a generic noise-aware aggregation method that can be paired with any federated strategy to mitigate the impact of contributions from noisy clients.

A. Problem Definition

In a typical FL scenario, a total of K clients collaboratively train a machine learning model by optimizing a global objective function [1]. To formulate this setting, consider K disjoint local datasets by clients, where each client indexed by k holds a private dataset $D_k = \{(x_i, y_i)\}_{i=1}^{|D_k|}$, with each pair (x_i, y_i) representing a data sample such that $x_i \in \mathcal{X}$ as the input and $y_i \in \mathcal{Y}$ as the corresponding label. The objective of the clients is to minimize the loss $\mathcal{L}(\theta)$ aggregated from the individual loss of each local client $\ell(\theta, D_k)$ with respect to the aggregated server model parameters θ :

$$\begin{aligned} \min_{\theta} \mathcal{L}(\theta) &= \min_{\theta} \frac{1}{\sum_{k \in [K]} |D_k|} \sum_{k \in [K]} \sum_{(x_i, y_i) \in D_k} \ell(\theta; \{(x_i, y_i)\}) \\ &= \min_{\theta} \sum_{k=1}^K w_k \ell(\theta; D_k) \end{aligned} \quad (1)$$

where w_k denotes the weight assigned to the k -th client's loss function. Here, we take w_k proportional to the size of local data D_k of client k such that $w_k = |D_k| / \sum_{k=1}^K |D_k|$, similar to that of [2].

Noisy Federated Learning We focus on a noisy federated learning setting where perturbations are present in the input space \mathcal{X} , that often occur due to the unexpected variability in data collection or physical conditions and other environmental factors [34]–[36].

Consider an FL setup consisting of K participating clients, where M out of K clients' local datasets are partially contaminated by some noise, whose function is defined as $\eta(\cdot) : \mathcal{X} \mapsto \mathcal{X}$. Conversely, the local datasets of the remaining N clients, where $N = K - M$, are unaffected. To simplify the exposure, we index the noisy clients by m and clean clients by n , and denote the respective client data by D_m and D_n , where $\{\cup_m D_m\} \cup \{\cup_n D_n\} = \cup_k D_k$. We assume that the local dataset of each noisy client m can be further split disjointly into clean data D_m^{clean} and noisy data D_m^{noisy} such that $D_m^{\text{clean}} \cup D_m^{\text{noisy}} = D_m$ and $D_m^{\text{clean}} \cap D_m^{\text{noisy}} = \emptyset$. In such noisy federated learning setting, the global objective can be expressed as:

$$\begin{aligned} \min_{\theta} \mathcal{L}(\theta) &= \min_{\theta} \sum_{\substack{n \in [K] \\ \text{s.t. } D_n \text{ is clean}}} w_n \ell(\theta; D_n) + \sum_{\substack{m \in [K] \\ \text{s.t. } D_m \text{ is noisy}}} w_m \ell(\theta; D_m) \\ &= \min_{\theta} \sum_{\substack{n \in [K] \\ \text{s.t. } D_n \text{ is clean}}} w_n \ell(\theta; D_n) \\ &\quad + \sum_{\substack{m \in [K] \\ \text{s.t. } D_m \text{ is noisy}}} w_m \left(\frac{|D_m^{\text{clean}}|}{|D_m|} \ell(\theta; D_m^{\text{clean}}) \right. \\ &\quad \left. + \frac{|D_m^{\text{noisy}}|}{|D_m|} \ell(\theta; D_m^{\text{noisy}}) \right). \end{aligned} \quad (2)$$



Fig. 3: Illustration of patch-based data corruption.

Noisy Input Data In decentralized data collection, input corruption typically occurs on the client side, influenced by varying degrees of real-world noise and varying degrees of corruption from weak to strong. To address this, we propose a definition for noisy FL data that accounts for the intensity and level of noise in each client's local datasets.

Consider the partition of a global training dataset D into K local datasets D_k for $k \in [K]$. This partitioning follows either an Independent and Identically Distributed (IID) or a non-IID setting [2]. Within this set of K clients, we randomly pick up M clients as the noisy clients, in which the local datasets of these clients are contaminated by input noise, such as Gaussian blur, contrast, defocus blur (see the extended list in the Section IV-A). Consequently, the local dataset of m -th noisy client D_m is transformed into noisy local data by applying a randomly selected transformation τ from all the data transformation \mathbf{T} with a specific severity level $\xi \in \{low, medium, high\}$. This can be expressed as:

$$D_m^{\text{noisy}} = \{ \eta(x, \tau, \xi) \mid x \in D_m, \tau \in \mathbf{T} \}. \quad (3)$$

Thus, we define the noise level of client m as $NL_m = \frac{|D_m^{\text{noisy}}|}{|D_m|}$, which represents the fraction of noisy corrupted data samples

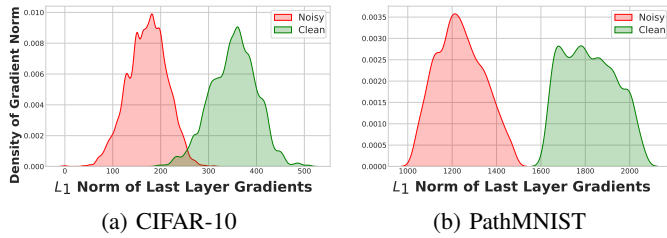


Fig. 4: Gradient norm density during federated training (15 clean, 5 noisy clients, $NL=100\%$) using mean aggregation.

over the entire local data of client m . We further introduce image noise severity to quantify the extent of distortion in digital images. Severity levels are scaled from low to high, indicating the intensity of the noise. An illustration of different levels of image noise severity is shown in Figure 2.

In addition to the image distortion, we further introduce patch-based image corruption to simulate the condition when the nullified data is injected into the client data. We term this injection of nullified images as *patch-based corruption*. We characterize these corruptions in the following three forms (but note that many others could exist): black patches, patches with Gaussian noise, and generative patches (we use StyleGANv2) [37], illustrated in Figure 3. Here, specific data samples in the client’s local dataset are substituted with such patches, while the corresponding labels remain unchanged, e.g., a random generative patch has an object label assigned to it.

B. Single-Interaction Discovery of Noisy Clients

Unlike centralized learning, where the server has direct access to inspect all the data, FL presents a unique challenge due to the decentralized and private nature of the data. In FL, clients contribute to the server by sharing their model parameters rather than raw data [1], [2], making it impossible to visually verify the quality of data. To address this issue, we propose a novel approach that can discover the noisy client with only a single interaction (or in a one-shot manner) using only a local model. Specifically, we identify the noisy clients by evaluating the gradient norms [8], [38] calculated from local data during the initial training round without compromising the client’s data confidentiality.

The challenge in training deep neural networks on input (such as images) with noise and distortions is that the model can easily be affected by the perturbations, thus producing overconfident predictions on distorted input data samples [6], [12], [13], [16] or leading to poor generalization. Motivated by the findings of [7], we empirically discover that a client’s local model trained on distorted inputs exhibits distinct behavior in its gradient space compared to a model trained on clean inputs (see Figure 4). In other words, the magnitude of gradients, when trained on clean or corrupted data, is indicative of variation in the model’s learning process. The gradient norm effectively encapsulates the relationship between input features and the model’s output. Intuitively, we propose a single-interaction method to estimate data distortion by evaluating the gradient norm of the softmax cross-entropy loss \mathcal{L} during the backpropagation phase of local model training that we get

at no extra cost. Given the model parameterized by θ , we define the loss of the input batch data \mathbf{x} with its ground true labels \mathbf{y} as:

$$\mathcal{L}(\theta; \mathbf{x}, \mathbf{y}) = -\log \left(\frac{e^{f_{\mathbf{y}}(\theta; \mathbf{x})/T}}{\sum_{c=1}^C e^{f_c(\theta; \mathbf{x})/T}} \right) \quad (4)$$

where the $f(\cdot)$ signifies the model’s output function, $f_{\mathbf{y}}(\theta; \mathbf{x})$ is the output logit from the model f for the ground true label \mathbf{y} given the input \mathbf{x} , and $f_c(\theta; \mathbf{x})$ is the output logit from the neural network for class c . The gradient norm of the input batch data \mathbf{x} over C classes is defined as follows:

$$g(\mathbf{x}) = \|\nabla_{\theta} \mathcal{L}(\theta; \mathbf{x}, \mathbf{y})\|_p \quad (5)$$

The term $g(\mathbf{x})$ refers to the gradient norm of the input batch data \mathbf{x} . Considering the k -th client within a set of clients, the overall local dataset of client D_k is split into B mini-batches. Hence, we compute the gradient norms for the entire local dataset D_k , expressed as $\{g(\mathbf{x}_b)\}_{b=1}^B$. After the initial round of local training, the server aggregates the gradient norm results from each of the N clients, forming a score vector by computing variance. By applying the kernel density estimation on the aggregated gradient norm vectors from the clients, we observe a distinct separation in the distributions of clean and noisy clients in Figure 4. Subsequently, we apply the K-means clustering on the set of all client gradient norm vectors to distinguish two clusters, as illustrated in Figure 1. We then compute the centroid of each cluster, categorizing the one with the higher centroid value as ‘clean’ and its counterpart as ‘noisy’ inspired by the findings of [8]. We examine the outcomes of client clustering within the FedNS framework. During the initial training round, the server categorizes all clients into either clean or noisy clusters based on gradient norms. We then visualize 9 images sampled from both clean and noisy clusters. As shown in Figure 5, the images from the noisy cluster are contaminated by a mixture of noise, whereas those from the clean cluster remain intact.

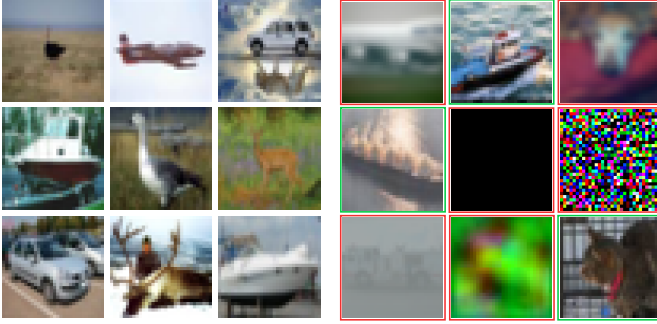
A key advantage of our proposed method is its implementation at the beginning of the federated training process, eliminating the need for additional communication rounds. By utilizing only the aggregated gradient norms from each client, our approach is directly usable (as a plug-in) with various FL strategies. Our method incorporates only the gradient norm (i.e., a scalar value) along with standard model parameters during the aggregation phase, thereby not increasing communication overhead. Moreover, this method enhances security by requiring clients to share only scalar values, significantly reducing the risk of sensitive data leakage.

C. Federated Noise Sifting

In FL, the formulation of the global model normally involves the aggregation of local models from each client, weighted based on the total count of data samples in their respective local datasets [1], [2]. When applying an average aggregation strategy when clients’ data follows the IID setting, the contribution of each client’s model parameters to the server

TABLE I: Comparison of average accuracy across three independent runs for different datasets under clean and noisy client data scenarios. For the noisy data scenario, we consider 5 clean clients and 15 noisy clients with 100% noise level. Models are trained with FedAvg.

Data	CIFAR10		CIFAR100		PathMNIST		FMNIST		EuroSAT		Tiny-ImageNet	
	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID
Clean	90.14%	85.52%	64.79%	62.36%	87.74%	82.55%	92.34%	89.37%	94.72%	95.12%	53.26%	52.88%
Noisy	78.62%	73.51%	44.58%	42.10%	54.80%	52.14%	88.14%	84.67%	67.39%	75.06%	24.32%	22.90%



(a) Samples from clean clients (b) Samples from noisy clients

Fig. 5: Qualitative analysis of clean and noisy CIFAR-10 samples identified by FedNS. Nine images from each cluster are shown. In Figure 5b, noisy samples exhibit mixed corruption. Red and green grids denote noisy and clean images, respectively.

is equivalent. Consequently, in scenarios of noisy federated learning, the contribution of noisy clients can seriously degrade the generalization quality of the server-side aggregated model, as detailed in Table I.

Normally, the optimization function of a standard noisy FL problem is to minimize the loss from both noisy and clean clients defined in Equation 2, where the model parameters from each client’s model are weighted by the amount of data it has divided by the total number of data samples in all clients. To mitigate the detrimental effect of noisy clients, we propose a noise-sensitive weights aggregation strategy that is resilient to data corruption during the federated training process, termed FedNS. Specifically, we introduce the weighting factor α and β to control the contribution of model updates from clean and noisy clients, respectively. Given a set of client updates and their corresponding weights, the global parameter θ is updated by aggregating the weighted updates from all clients, including both clean and noisy ones. Considering θ_g^t as the global parameter at the training round t , K total clients including N clean clients and M noisy clients. Let w_k be the weight and θ_k be the model parameter of client k . Thus, We formulate the proposed global parameter update as follows:

$$\theta_g^t \leftarrow \theta_g^{t-1} - \eta \left(\sum_{\substack{n \in [K] \\ \text{s.t. } D_n \text{ is clean}}} w_n \alpha \nabla \ell(\theta_n^t; D_n) + \sum_{\substack{m \in [K] \\ \text{s.t. } D_m \text{ is noisy}}} w_m \beta \nabla \ell(\theta_m^t; D_m) \right) \quad (6)$$

where $\nabla \ell(\theta_n^t; D_n)$ represents the gradient of the loss function for the clean client n at training round t , while $\nabla \ell(\theta_m^t; D_m)$

denotes the corresponding gradient for the noisy client m in the same training round. η denotes the learning rate. The coefficients α and β are utilized to proportionally scale the model update from clean and noisy clients, respectively. Concisely, our approach involves assigning a higher weighting factor to clean clients and a reduced one to noisy clients to lower their contributions to the overall global model. We provide the complete FedNS algorithm in Algorithm 1.

IV. EXPERIMENTS

A. Datasets, Models, and Configuration

Datasets. To demonstrate the efficacy of FedNS, we develop the noisy datasets benchmark based on [39]. we consider the image classification task and test our approach across various datasets, including CIFAR10/100 [40], Path-MNIST [41], Fashion-MNIST [42], EuroSAT [43], and Tiny-ImageNet [44]. These datasets are further applied data corruptions that are defined in Section III-A to create the noisy input for FL. Specifically, we categorize data corruption into distortion and patch-based noise. In practice, distortion corruption consists of noise addition (e.g., Gaussian noise, shot noise, impulse noise, pixelated noise), blur (e.g., defocus blur, glass blur, motion blur, Gaussian blur), photometric distortions (e.g., brightness, contrast, saturate), synthetic distortions (e.g., snow, fog, elastic, spatter), and compression artifact (e.g., JPEG compression). Moreover, we further evaluate our method on patch-based obfuscation noise, including black patch, Gaussian noise patch, and StyleGAN-based patch [37].

Baselines and Implementation. We apply FedNS in conjunction with four widely used FL strategies to evaluate its effectiveness, namely FedAvg [2], FedProx [9], FedTrimmedAvg [10], and FedNova [11]. We employ ResNet-18 as the main model architecture, utilizing the mini-batch SGD as the universal local optimizer for all FL strategies. Moreover, we further evaluate FedNS on a different neural architecture ConvMixer-256/8 in Table III. The optimizer is characterized by a learning rate of 0.01, an SGD momentum of 0.9, and the weight decay is set to 10^{-4} . We set the number of local training epochs to 5 and the global communication rounds to 150 across all datasets. We consider a setup with $N = 20$ clients for our experiments unless mentioned otherwise. This choice aligns with standard practices in FL and accommodates our computational limitations. The training set is distributed under both IID and non-IID settings. For the main experiments, we divide the client set into 15 noisy clients and 5 clean clients, with full client participation $r_p = 1.0$. We compute L_1 -norm of the gradient of the last layer for noisy client detection in all the cases, see Table IX for comparison between the effectiveness of L_1 and L_2 norms, as well as the impact of

Algorithm 1 Federated Noise Sifting (FedNS): Our method can be easily plugged into various FL strategies like the standard Federated Averaging. N is local clients, T is the total training rounds, E is the epoch of local training, and η is the learning rate. At each round t , the server will sample S_t clients that can be divided into n clean clients and m noisy clients. (**FedAvg**, **FedNS**)

Server executes:

```

1: Initialization: The server initializes the global model with the model parameter  $\theta_g$ 
2: for  $t = 0, \dots, T - 1$  do
3:    $S_t \leftarrow$  Sample a subset of clients
4:    $\theta_i^t \leftarrow \theta_g^t$ 
5:   for each client  $i \in S_t$  in parallel do
6:      $\theta_i^{t+1} \leftarrow$  ClientUpdate( $i, \theta_i^t, t$ ) ▷ Update client  $i$  and compute gradient norm
7:   end for
8:   if  $t = 0$  then ▷ Only execute at the initial training round
9:     for each client  $i \in S_t$  in parallel do
10:       $g_i \leftarrow$  GradNorm( $i, \theta_i^t, t$ ) ▷ Calculate gradient norm for each client  $i$ 
11:    end for
12:     $\mathcal{C}_{clean}, \mathcal{C}_{noisy} \leftarrow C(\{g_i\})$  ▷ Clustering client  $i$  into clean or noisy based on  $g_i$ 
13:     $\alpha_t, \beta_t \leftarrow W(\mathcal{C}_{clean}, \mathcal{C}_{noisy}, i)$  ▷ Assign weighting factors to each cluster
14:  else
15:     $\alpha_t \leftarrow \alpha_{t-1}$  and  $\beta_t \leftarrow \beta_{t-1}$ 
16:  end if
17:   $\theta_g^{t+1} \leftarrow \theta_g^t - \eta \sum_{i \in S_t} \mathbf{w}_i \nabla \ell(\theta_i^t)$  (FedAvg)
18:   $\theta_g^{t+1} \leftarrow \theta_g^t - \eta \left( \sum_{i \in n} \mathbf{w}_i \alpha_t \nabla \ell(\theta_i^t) + \sum_{j \in m} \mathbf{w}_j \beta_t \nabla \tilde{\ell}(\theta_j^t) \right)$  (FedAvg+FedNS)
19: end for
Client executes:
20: Local update: ClientUpdate( $i, \theta_i, t$ ) ▷ Client  $i$  executes update at round  $t$ 
21: for  $e = 0, \dots, E - 1$  do
22:   Update  $\theta_i$ 
23: end for
24: return  $\theta_i$ 
25: Compute gradient norm: GradNorm( $i, \theta_i, t$ )
26: if  $t = 0$  then
27:    $g_i \leftarrow \|\nabla f(\theta_i)\|$  ▷ Compute gradient norm at the beginning round  $t_0$ 
28: end if
29: return  $g_i$ 

```

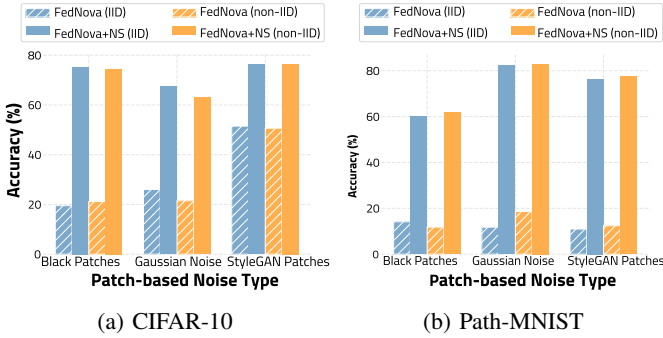


Fig. 6: Comparative results of FedNS in dealing with patch-based noise types on CIFAR-10 and Patch-MNIST. For consistency, we utilize the same noise setup as in Table II and conduct the experiments on both IID and non-IID settings.

batch size on the detection performance. We conducted all experiments on NVIDIA A10 GPUs.

B. Results and Ablation Studies

Participation of noisy clients deteriorates the performance of the global model. To validate the efficacy of our proposed method, we first conduct an experiment for model training with clean and noisy input across all the datasets and utilize the same noise configuration for our further empirical evaluation. With this, we aim to evaluate the upper-bound performance that can be achieved when learning from a mixture of noisy and clean clients. Table I presents the

comparative results of average accuracy for all considered datasets. We focus on three specific distortions (i.e., defocus blur, Gaussian blur, contrast) due to their significant impact on degrading the model’s generalization capability to simulate the worst case in noisy FL. For the generation of distorted data used in the experiments summarized in Table II, each noisy sample was produced by randomly selecting a distortion type with the configuration characterized by the noise severity level $\xi = high$. We set the noise level to $NL_m = 100\%$ for every client m for the experiments of both Table II and Figure 6. We see the participation of noisy clients leads to a significant degradation in the model’s generalization capability across all tasks, indicating the detrimental impact of noisy data in the FL environment. Furthermore, due to the inherent lack of visibility into the data from these federated clients, the resultant global model tends to be of low quality, and it may become challenging in a real-world setting to identify the underlying reasons for its poor performance.

FedNS significantly improves standard federated aggregation methods. We investigate the robustness of our proposed method by applying FedNS on six image datasets with different settings under the noisy scenario. As shown in Table II, the performance of all aggregation methods exhibits a general trend of improvement by simply plugging FedNS into the considered strategies. In particular, we consider the worst-

TABLE II: Comparison of top-1 accuracy across datasets in IID and Non-IID settings. We evaluate the performance of FedNS with various federated aggregation methods for learning under the noisy environment.

Methods	CIFAR-10		CIFAR-100		PathMNIST		FMNIST		EuroSAT		Tiny-ImageNet	
	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID
FedAvg [2]	78.62%	73.51%	44.58%	42.10%	54.80%	52.14%	88.14%	84.67%	67.39%	75.06%	24.32%	22.90%
+ NS (Ours)	81.67%	78.44%	48.14%	45.94%	63.89%	62.92%	89.61%	88.53%	78.22%	80.12%	27.85%	25.93%
FedProx [9]	79.89%	78.13%	46.75%	45.17%	57.28%	56.27%	87.15%	86.96%	70.83%	76.64%	24.90%	23.76%
+ NS (Ours)	82.31%	81.18%	48.27%	46.80%	60.18%	63.11%	89.12%	87.48%	76.94%	81.20%	26.48%	25.98%
FedTrimmedAvg [10]	78.92%	77.24%	41.81%	41.25%	56.34%	54.50%	90.09%	89.95%	68.30%	74.39%	16.97%	15.48%
+ NS (Ours)	82.63%	82.47%	49.11%	48.32%	64.27%	63.04%	90.29%	91.57%	83.81%	80.50%	29.43%	27.46%
FedNova [11]	81.45%	82.16%	49.48%	48.24%	55.36%	51.04%	90.65%	89.68%	73.54%	66.29%	28.62%	27.24%
+ NS (Ours)	88.65%	88.34%	59.19%	59.17%	80.82%	81.89%	90.57%	91.50%	93.31%	92.70%	48.50%	46.16%

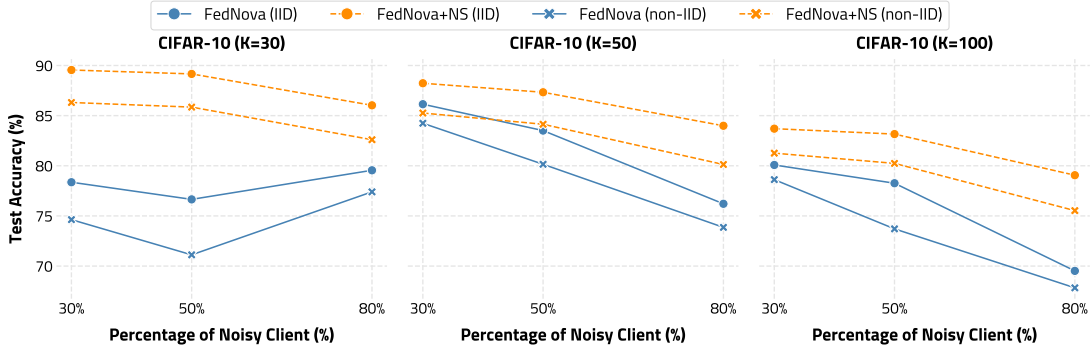


Fig. 7: Impact of the number of clients on model performance under different levels of client data distribution noise with FedNova.

case with heterogeneous data setting in Table II, where 15 out of 20 noisy clients participate in the federated training with high noise severity and 100% noise level. Adding FedNS to FL strategies yields better overall performance among all the datasets, especially for some vulnerable datasets (e.g., Path-MNIST) that are sensitive to data corruption. Additionally, we demonstrate the efficacy of FedNS in dealing with patch-based noise on CIFAR-10 and Path-MNIST as presented in Figure 6. From Figure 6, we observe that FedNS consistently boosts the performance of the aggregation method across all the datasets and settings. This further shows that FedNS is capable of handling various types of noises a model can encounter in a real-world setting.

Impact of noisy client distributions on performance across varying numbers of clients. We evaluate FedNS under the influence of noise client percentages in Figure 7 with different amounts of clients at scale, where K denotes the number of clients. We select K from $\{30, 50, 100\}$ and set the noisy client percentage as 30%, 50%, 80%, respectively. As shown in the figure, the results exhibit a consistent pattern of variation in model performance across different scenarios. Specifically, as the percentage of noisy clients increases, there is a gradual decline in test accuracy for both IID and non-IID distributions. However, FedNova+NS (IID) consistently outperforms the other methods, maintaining higher accuracy levels across all settings. This demonstrates the robustness and effectiveness of the FedNova+NS method in mitigating the adverse effects of noisy clients, especially as the number of clients increases.

Impact of Client Participation Rate (r_p). Next, we examine how the client participation rate (r_p) affects the performance of FedNS. We conduct experiments on CIFAR-10 and Path-MNIST, maintaining a fixed noise configuration across all settings. We vary r_p to simulate different device participation patterns, choosing values from $r_p \in \{0.2, 0.5, 1.0\}$, where $r_p = 1.0$ denotes full client participation. In practice, the server ensures full client participation during the first training round to identify noisy clients. Figure 8 provides the results. We observe that incorporating FedNS significantly enhances performance across all client participation patterns. Our experiments suggest that FedNS remains robust to variations in client availability and data quality.

Robustness of FedNS on mixed noise conditions. We investigate the robustness of FedNS under complex noise conditions that involve a combination of different noise types, including the distortions and patch-based noises as described in Section IV-A. Table IV shows the performance gains achieved by FedNS are particularly significant in high-variance datasets such as Path-MNIST and Tiny-ImageNet, where FedNova+NS demonstrates substantial improvements over standard aggregation. Notably, FedNS consistently enhances the global model's performance across all evaluated datasets, showcasing its robustness in handling intricate noise scenarios. This highlights the effectiveness of FedNS in mitigating the impact of complex noise conditions, where decentralized data may be subject to various types of distortions at the same time.

Evaluation on Weight Factor (β) of Noisy Client. We further investigate the effect of weight factor β on over-

TABLE III: Top-1 accuracy of the ConvMixer-256/8 model using FedNova aggregation strategy in a high-noise environment.

Methods	CIFAR-10		CIFAR-100		PathMNIST		FMNIST		EuroSAT		Tiny-ImageNet	
	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID
FedNova	71.76%	64.50%	37.31%	36.23%	54.15%	49.75%	84.43%	86.31%	61.70%	56.28%	21.04%	19.43%
+ NS (Ours)	77.05%	70.51%	44.17%	43.07%	79.76%	80.82%	85.68%	86.71%	88.59%	87.80%	33.66%	31.80%

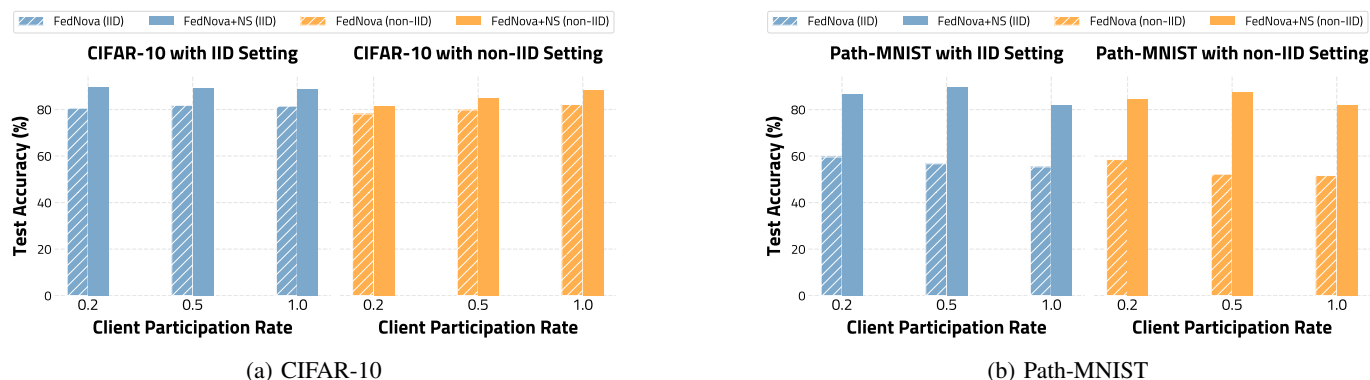


Fig. 8: Ablation on the impact of different client participation rates (r_p). Evaluation on the CIFAR-10 and Path-MNIST datasets, with varying r_p selections. The results compare model generalization performance with and without FedNS across different experimental setups.

TABLE IV: Top-1 accuracy of the FedNova aggregation method (+FedNS) in an IID setting on selected datasets when noisy clients' data is a mixture of all noise types.

Noise Setting	Clean		Noisy	
	FedNova	FedNova	FedNova	FedNova+NS
CIFAR-10	90.14%	80.10%	85.83%	(\uparrow 5.73%)
CIFAR-100	64.79%	45.28%	51.47%	(\uparrow 6.19%)
Path-MNIST	92.34%	65.19%	87.81%	(\uparrow 22.62%)
Tiny-ImageNet	53.26%	29.26%	43.30%	(\uparrow 14.04%)

all performance in FedNS. The weight factor β controls the weights of aggregated noisy client local models, as described in Equation 6. We considered various values of $\beta \in \{0, 0.1, 0.3, 0.5, 0.7, 1.0\}$, where $\beta = 0$ signifies the exclusion of noisy model weights, and $\beta = 1.0$ indicates the direct aggregation of model weights. As shown in Table V, the setting of $\beta = 0.3$ yields the best performance, suggesting an optimal trade-off. Interestingly, the exclusion of noisy model weights ($\beta = 0.0$) leads to a degradation in the generalization capability of the global model. This observation suggests that incorporating mitigated noisy data enhances the robustness of the global model.

TABLE V: Ablation on the weight factor β for noisy clients. We evaluate this parameter on the CIFAR-10 dataset with IID and non-IID settings, and the weight factor of clean clients α is set to 2.0.

Setting	Weight Factor β of Noisy Client					
	0.0	0.1	0.3	0.5	0.7	1.0
IID	82.82	87.17	88.65	87.53	86.28	84.33
non-IID	75.31	82.49	88.34	84.05	83.44	81.27

FedNS on Real-world Human Annotation Errors. In this experiment, we extend our investigation to assess the efficacy of FedNS in addressing real-world data quality issues, specifically human annotation errors. While this work focuses on mitigating data corruption in the input space, we identify that FedNS can effectively handle label noise - a well-studied problem in noisy FL. Label noise occurs when data labels are incorrectly assigned while the input features remain unaltered. Specifically, we assessed FedNS on CIFAR-10/100N, two benchmark datasets featuring real-world noisy labels resulting from human annotation errors [45]. We employed FedAvg and FedNova as baseline approaches, adhering to the training configuration outlined in Section IV-A. From Table VI, we observe that FedNS consistently improved the performance across all the experiments. Our findings demonstrate that FedNS not only excels in mitigating input space corruption but also shows promising results in handling label noise. The ability to address both input space corruption and label noise positions FedNS as a valuable tool for practitioners dealing with real-world datasets, where multiple types of data imperfections may coexist.

TABLE VI: Performance of FedNS on datasets with noisy labels. We evaluate FedNS on CIFAR-10/100N [45] datasets.

Methods	CIFAR-10N		CIFAR-100N	
	IID	Non-IID	IID	Non-IID
FedAvg	76.06%	69.52%	52.61%	52.57%
+ NS (Ours)	78.87%	71.14%	54.06%	53.85%
FedNova	76.06%	69.26%	53.32%	52.41%
+ NS (Ours)	84.38%	78.65%	55.54%	54.80%

Impact of Initial-Round Client Participation on FedNS Performance. In FL, the assumption that all clients must be warmed up and participate from the very first training

round may not always hold true, as some clients may only participate in later rounds. To address this, we explore the scenario where only a subset of clients participates in the initial round. Instead of collecting gradient norms solely in the first round, we iteratively apply FedNS until all clients have been engaged. We evaluate this approach by setting the first-round participation rates to 10%, 50%, and 100%, with 100% representing full client participation. The results, presented in Table VII, highlight the robustness of FedNS across varying levels of initial-round client participation.

TABLE VII: FedNS performance on CIFAR-10 and CIFAR-100: Classification accuracy with 10%, 50%, and 100% client participation in the first round, i.e., partially warmed-up clients.

Methods	CIFAR-10			CIFAR-100		
	10%	50%	100%	10%	50%	100%
First-Round Participation						
FedNova	80.46%	81.14%	81.45%	48.63%	48.46%	49.48%
+ NS (Ours)	86.13%	86.76%	88.65%	54.13%	54.27%	59.19%

FedNS on Different Noise Level (NL) of Client Data.

In Table VIII, we examine the performance of FedNS across various noise configurations. Specifically, we consider noise levels $NL \in \{50\%, 80\%, 100\%\}$ and noise severity $\xi \in \{Medium, High\}$. These results demonstrate that FedNS substantially enhances model generalization under high noise conditions, especially when clients' data is completely corrupted (i.e., 100% noise level and high severity). Conversely, in scenarios with milder noise where the impact on federated models is minimal, the implementation of FedNS does not negatively affect the generalization process. Hence, we limit our experiments to noise levels at or above 50%, as noise levels below this threshold have negligible impact on model performance.

TABLE VIII: Ablation for different noise level and noise severity with IID setting on CIFAR-10/100 datasets. We employ FedNova as the aggregation method and ablate the noise level and noise severity.

Noise Level(NL)	$NL = 50\%$		$NL = 80\%$		$NL = 100\%$	
	Medium	High	Medium	High	Medium	High
FedNova	89.95%	89.02%	90.02%	88.90%	87.95%	81.45%
+ NS (Ours)	90.56%	90.11%	90.35%	90.12%	90.35%	88.65%

(a) CIFAR-10

Noise Level(NL)	$NL = 50\%$		$NL = 80\%$		$NL = 100\%$	
	Medium	High	Medium	High	Medium	High
FedNova	65.51%	62.70%	64.18%	60.96%	60.85%	49.48%
+ NS (Ours)	65.03%	62.83%	65.17%	62.14%	62.76%	59.19%

(b) CIFAR-100

Hyperparameters Selections for Noisy Clients Detection.

One of the essential components of FedNS is using gradient norms to identify noisy clients. We investigate several factors that may affect the performance of detecting noisy clients, specifically the selection of the L_p -norm and the batch size of gradient norms for clustering. This ablation experiment is performed on the CIFAR-10 and Path-MNIST datasets, as presented in Table IX.

The comparison between the L_1 -norm and L_2 -norm shows that the L_1 -norm generally outperforms higher-order norms.

Moreover, we evaluate the impact of batch size by setting the mini-batch size to range from 1 to 128. Our experiments suggest that a larger batch size effectively reduces the variance of the gradient estimates. The results, shown in Table IX, indicate that a batch size selection around 16 \sim 64 yields better performance across all settings.

TABLE IX: Hyperparameter impact on noisy client detection: CIFAR-10 and Path-MNIST, FedNova, 5 clean + 15 noisy clients (100% noise), IID, full participation in first round, 5 local epochs. Values show noisy client detection accuracy.

L_p -Norm	Batch Size of Gradient Norm					L_p -Norm	Batch Size of Gradient Norm				
	1	16	32	64	128		1	16	32	64	128
L_1 -Norm	0.85	1.00	1.00	0.95	0.90	L_1 -Norm	0.75	1.00	1.00	1.00	0.90
L_2 -Norm	0.90	0.95	0.90	0.90	0.95	L_2 -Norm	0.60	0.75	0.85	0.90	0.90

(a) CIFAR-10

(b) Path-MNIST

Evaluating FedNS with alternative model architecture.

In this experiment, we investigate the robustness of FedNS when applied to a different model architecture. We employ the ConvMixer-256/8 [46] model and train it using FedNova on a range of datasets. The noise configuration remains consistent with the details provided in Section IV-B, and we evaluate the model's performance under both IID and non-IID settings. As shown in Table III, the federated aggregation method, when paired with FedNS, achieves enhanced performance across all considered datasets. These improvements, observed consistently across tasks, demonstrate the adaptability and effectiveness of FedNS when integrated with alternative architectural paradigms. The results highlight the versatility of our approach in enhancing the performance of different federated model regardless of the neural architecture, further emphasizing its potential as a robust and flexible method for FL.

V. CONCLUSION

In this paper, we study the training of the federated neural networks within a noisy environment in which the input space of client data is corrupted. We propose to utilize the gradient norm of local training updates to discover the noisy clients. We then propose FedNS, an effective method designed to integrate with diverse FL aggregation methods to mitigate the impact of model updates from noisy clients. Our experimental results on several benchmark datasets show that FedNS is robust against data perturbation, which can significantly boost the generalization of federated models. We further present extensive ablation studies to provide a better understanding of FedNS. Our findings highlight the importance for FL approaches that prioritize robustness when training models on decentralized data. By focusing on this aspect, we can enhance the reliability and stability of FL in a real-world setting, where data quality may vary significantly across participating clients. Future work will investigate FedNS's efficacy in the language modeling domain and learning from multi-modal noisy data in a decentralized setting.

ACKNOWLEDGMENT

This work was partially supported by the NGF AiNed Fellowship Grant of A.S.

REFERENCES

- [1] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.
- [2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017.
- [3] V. Gudivada, A. Apon, and J. Ding, "Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations," *International Journal on Advances in Software*, 2017.
- [4] L. Budach, M. Feuerpfeil, N. Ihde, A. Nathansen, N. Noack, H. Patzlaff, F. Naumann, and H. Harmouch, "The effects of data quality on machine learning performance," *arXiv preprint arXiv:2207.14529*, 2022.
- [5] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in machine learning*, 2021.
- [6] S. Dodge and L. Karam, "A study and comparison of human and deep learning recognition performance under visual distortions," in *2017 26th international conference on computer communication and networks (ICCCN)*. IEEE, 2017.
- [7] N. Gupta, S. Mujumdar, H. Patel, S. Masuda, N. Panwar, S. Bandyopadhyay, S. Mehta, S. Guttala, S. Afzal, R. Sharma Mittal *et al.*, "Data quality for machine learning tasks," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021.
- [8] R. Huang, A. Geng, and Y. Li, "On the importance of gradients for detecting distributional shifts in the wild," *Advances in Neural Information Processing Systems*, 2021.
- [9] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, 2020.
- [10] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. Pmlr, 2018.
- [11] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, 2020.
- [12] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *2016 eighth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2016.
- [13] R. Geirhos, C. R. M. Temme, J. Rauber, H. H. Schütt, M. Bethge, and F. A. Wichmann, "Generalisation in humans and deep neural networks," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018.
- [14] T. S. Borkar and L. J. Karam, "Deepcorrect: Correcting dnn models against image distortions," *IEEE Transactions on Image Processing*, 2019.
- [15] D. Liang, X. Gao, W. Lu, and L. He, "Deep multi-label learning for image distortion identification," *Signal processing*, 2020.
- [16] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the robustness of deep neural networks via stability training," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [17] Y. Zhou, S. Song, and N.-M. Cheung, "On classification of distorted images with deep convolutional neural networks," in *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017.
- [18] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [19] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Learning to learn from noisy labeled data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.
- [20] A. Ghosh, H. Kumar, and P. S. Sastry, "Robust loss functions under label noise for deep neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, 2017.
- [21] B. Han, Q. Yao, T. Liu, G. Niu, I. W. Tsang, J. T. Kwok, and M. Sugiyama, "A survey of label-noise representation learning: Past, present and future," *arXiv preprint arXiv:2011.04406*, 2020.
- [22] Y. Chen, X. Yang, X. Qin, H. Yu, P. Chan, and Z. Shen, "Dealing with label quality disparity in federated learning," *Federated Learning: Privacy and Incentive*, 2020.
- [23] M. Yang, H. Qian, X. Wang, Y. Zhou, and H. Zhu, "Client selection for federated learning with label noise," *IEEE Transactions on Vehicular Technology*, pp. 2193–2197, 2021.
- [24] X. Fang and M. Ye, "Robust federated learning with noisy and heterogeneous clients," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
- [25] J. Xu, Z. Chen, T. Q. Quek, and 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.
- [26] S. Yang, H. Park, J. Byun, and C. Kim, "Robust federated learning with noisy labels," *IEEE Intelligent Systems*, pp. 35–43, 2022.
- [27] J. Zhang, D. Lv, Q. Dai, F. Xin, and F. Dong, "Noise-aware local model training mechanism for federated learning," *ACM Transactions on Intelligent Systems and Technology*, pp. 1–22, 2023.
- [28] B. Zeng, X. Yang, Y. Chen, H. Yu, and Y. Zhang, "Clc: A consensus-based label correction approach in federated learning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, pp. 1–23, 2022.
- [29] V. Tsouvalas, A. Saeed, T. Ozecebi, and N. Meratnia, "Labeling chaos to learning harmony: Federated learning with noisy labels," *ACM Transactions on Intelligent Systems and Technology*, pp. 1–26, 2024.
- [30] T. Wang, J. Rausch, C. Zhang, R. Jia, and D. Song, "A principled approach to data valuation for federated learning," *Federated Learning: Privacy and Incentive*, 2020.
- [31] W. Li, S. Fu, F. Zhang, and Y. Pang, "Data valuation and detections in federated learning," *arXiv preprint arXiv:2311.05304*, 2023.
- [32] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*. PMLR, 2020.
- [33] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," *Advances in Neural Information Processing Systems*, 2020.
- [34] P. Singh and R. Shree, "A comparative study to noise models and image restoration techniques," *International Journal of Computer Applications*, 2016.
- [35] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: An overview," *Neural Networks*, 2020.
- [36] A. Villar-Corrales, F. Schirmacher, and C. Riess, "Deep learning architectural designs for super-resolution of noisy images," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.
- [37] M. Baradad Jurjo, J. Wulff, T. Wang, P. Isola, and A. Torralba, "Learning to see by looking at noise," *Advances in Neural Information Processing Systems*, 2021.
- [38] M. Paul, S. Ganguli, and G. K. Dziugaite, "Deep learning on a data diet: Finding important examples early in training," *Advances in Neural Information Processing Systems*, 2021.
- [39] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *Proceedings of the International Conference on Learning Representations*, 2019.
- [40] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [41] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification," *Scientific Data*, 2023.
- [42] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.
- [43] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.
- [44] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, 2015.
- [45] J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, and Y. Liu, "Learning with noisy labels revisited: A study using real-world human annotations," *arXiv preprint arXiv:2110.12088*, 2021.
- [46] A. Trockman and J. Z. Kolter, "Patches are all you need?" *arXiv preprint arXiv:2201.09792*, 2022.