A study about the performance of federated learning under out-of-distribution data groups

Minghao Wang*

College of Letters & Science, University of California, Santa Barbara, USA

Abstract. Federated learning (FL) is a promising machine learning paradigm that focuses on sending data to the client, then using the local unique data to train the local model. Only the model updates are sent back to be aggregated into a global model; this action can efficiently protect the data privacy and security. However, developing the FL model in the real world is still unreliable due to the problem that the real-world dataset always contains both desired in-distribution data and unexpected out-of-distribution (OOD) data. Current FL researchers typically use prefabricated artificial data sets to conduct a series of experiments and obtain positive results, yet most of these datasets are in-distribution data. In this work, we propose a light weight Client-Side OOD Filtering method tailored for real-world FL compatibility. This method integrates seamlessly with standard FL frameworks by adding a simple pre-processing step on the client side, without modifying server-side aggregation logic. The core design includes two key components: (1) A lightweight ID classifier that quantifies the alignment between local data and ID distribution with soft max probability. (2) An adaptive threshold mechanism that retains data with probability $\geqslant$ IID data for regional training and filters out low probability data.

1 Introduction

Federated learning has emerged as a transformative approach to protecting machine learning from privacy during the training process, which allows multiple clients to collaboratively train models without sharing their own raw data [1, 2]. The standard FL workflow involves iterative rounds of local training on client devices and has the aggregation of model updates on a central server [3]. This paradigm has found applications in various domains, including mobile keyboard prediction [4], hospital in different countries sharing their healthcare data [5], and many other systems [6]. Despite theoretical promise, real-world FL deployment faces significant challenges due to the inconsistency of the real-world data.

Particularly problematic is the presence of out-of-distribution data that does not align with the target task distribution [7]. In practical scenarios, client devices often collect data locally from multiple sources and environments, resulting in mixtures of relevant (ID) and irrelevant (OOD)data. For example, a federated image classification system might receive medical images from hospital clients but also may receive unrelated personal photos on mobile devices. Current FL research typically assumes the dataset they got is clean and well-

* Corresponding author: minghao_wang@ucsb.edu

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

constructed datasets that only contain ID data [8, 9]. This assumption does not hold in real-world settings where clients have varying data quality and relevance. The presence of OOD data can severely degrade FL performance through several mechanisms: (1) contamination of local updates with irrelevant signal, (2) introduction of conflicting gradients during aggregation, and (3) dilution of useful information with noise. Existing approaches to handling OOD data in FL focus primarily on server-side detection and filtering [10, 11] or robust aggregation techniques [12]. However, in real-world circumstances, these methods often require significant modifications to the FL framework, lack compatibility with existing systems, and may introduce additional privacy concerns by requiring more information sharing between the clients and the server, which obviously is a big disadvantage. To address these limitations, we propose a client-side OOD filtering approach that operates locally on each client device before training. Our method leverages a lightweight pre-trained ID classifier to assess data relevance and applies an adaptive threshold to filter out OOD samples. This approach maintains the privacy benefits of FL while improving model performance without requiring server-side modifications.

2 Related work

2.1 Federated Learning Foundations

Federated learning was formally introduced by McMahan et al. with the FedAvg algorithm [1], which remains the basis for most FL approaches. Subsequent research has addressed various challenges including statistical heterogeneity [13], system heterogeneity [14], and privacy concerns [15]. Several surveys provide comprehensive overviews of FL techniques and applications.

Federated Learning: I consider a federated learning system consisting of a central server and m decentralized clients. Each client $k \in \{1,2,\ldots,m\}$ holds a private local data set $D_k = \{(x_i^{(k)}, y_i^{(k)})\}_{i=1}^{n_k}$, where $x_i^{(k)} \in R^d$ is the input and $y_i^{(k)} \in Y = \{1,\ldots,C\}$ is the is the class label. The total dataset is $D = \bigcup_{k=1}^{m} D_k$, with total size $n = \sum_{k=1}^{m} n_k$. Let $\theta \in R^p$ denote the global model parameter. The local empirical risk on client k is defined as:

$$\mathscr{L}_k(\theta) = \frac{1}{n_k} \sum_{(x_i^{(k)}, y_i^{(k)}) \in \mathcal{D}_k} \ell\left(f_\theta(x_i^{(k)}), y_i^{(k)}\right), \tag{1}$$

where f_θ is the prediction function, and l is the loss function. The overall federated learning objective is to minimize the weighted global loss:

$$\min_{\theta \in \mathbb{R}^p} \mathscr{L}(\theta) := \sum_{k=1}^{m} \frac{n_k}{n} \cdot \mathscr{L}_k(\theta). \tag{2}$$

Federated Averaging: FedAvg is a widely-used method to optimize the above objective. In each communication round t, the server selects a subset $St \subseteq \{1, \ldots, m\}$ of clients and broadcasts the global model θt. Each selected client performs local training by running E epochs of stochastic gradient descent (SGD):

$$\theta_{t+1}^{(k)} = \theta_t - \eta \cdot \nabla \mathscr{L}_k(\theta_t), \tag{3}$$

where η is the local learning rate. The server then aggregates the updates:

$$\theta_{t+1} = \sum_{k \in \mathcal{S}_t} \frac{n_k}{\sum_{j \in \mathcal{S}_t} n_j} \cdot \theta_{t+1}^{(k)}. \tag{4}$$

2.2 Out-of-Distribution Detection

OOD detection has been extensively studied in centralized machine learning settings [16]. Common approaches include confidence-based methods using *Softmax* probabilities [17], distance-based methods [18], and generative approaches [19]. However, these methods typically assume access to the entire data set and centralized computation, making them unsuitable for FL settings where data remains distributed and computation must be efficient.

2.3 OOD Handling in Federated Learning

Recent work has begun addressing OOD challenges in FL. Chen et al. [10] proposed a server side OOD detection method based on analyzing update similarities. Zhang et al. [11] developed a robust aggregation scheme that weights client contributions based on data quality estimates. However, these approaches require significant server side computation and may introduce privacy risks by requiring additional information sharing. Client-side OOD filtering remains relatively unexplored. Our work differs from existing approaches by focusing on lightweight, local filtering that preserves privacy and compatibility with existing FL frameworks.

3 Methodology

3.1 Problem Formulation

Consider a federated learning system with K clients. Each client k possesses a local dataset D_k containing both in-distribution (ID) and out-of-distribution (OOD) data:

$$D_k = D_k^{ID} \cup D_k^{OOD}$$

The standard FL objective function is:

$$\min_{\theta} \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{N} \mathscr{L}_k(\theta; \mathcal{D}_k), \tag{5}$$

where $N = \sum_{j=1}^{K} |\mathcal{D}_j|$ and $\mathscr{L}_k$, is the local empirical risk.
My method transforms this to:

$$\min_{\theta} \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{N} \mathscr{L}_k(\theta; \mathcal{D}_k), \tag{6}$$

where $\widetilde{\mathcal{D}}_k$ is the filtered ID dataset and $\widetilde{N} = \sum_{j=1}^{K} |\widetilde{\mathcal{D}}_j|$.

3.2 Client-Side OOD Filtering Framework

3.2.1 Lightweight ID Classifier

We define a compact CNN architecture $f_{\theta_{ID}}$ with parameters:

$$\begin{aligned} \phi(x) &= CNN(x), \\ h(x) &= ReLU(W_1\phi(x) + b_1, \\ f_{\theta_{ID}}(x) &= W_2 h(x) + b_2. \end{aligned} \tag{7}$$

Trained with cross-entropy loss:

$$\mathscr{L}_{ID} = -\frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \sum_{c=1}^{C} y_{i,c} \log p_{i,c}\,, \tag{8}$$

where $p_{i,c} = softmax(f_{\theta_{ID}}(x_i))_c$.

3.2.2 OOD Scoring and Filtering

For each sample x, compute the maximum soft max probability:

$$s(x) = \max_{c\in\{1,\ldots,C\}} softmax\,(f_{\theta_{ID}}(x))_c, \tag{9}$$

apply threshold filtering:

$$\widetilde{\mathcal{D}}_k = \{(x,y) \in \mathcal{D}_k \mid s(x) \geq \tau\}. \tag{10}$$

3.3 Theoretical Analysis

3.3.1 Convergence Guarantees

Assume the loss function is L-smooth and μ-strongly convex. After T communication rounds, the expected optimality gap satisfies:

$$\mathbb{E}[\|\theta_T - \theta^*\|^2] \leq (1-\mu\eta)^T \|\theta_0 - \theta^*\|^2 + \frac{\sigma^2}{\mu\eta K}, \tag{11}$$

where σ^2 represents the variance reduction achieved by OOD filtering.

3.3.2 Error Decomposition

The generalization error can be decomposed as:

$$\mathscr{E} = \underbrace{\mathscr{E}_{ID}}_{ID\ error} + \underbrace{\mathscr{E}_{OOD}}_{OOD\ bias} + \underbrace{\mathscr{E}_{filter}}_{filtering\ error}\,. \tag{12}$$

Our method minimizes through effective filtering while maintaining bounded.

3.3.3 Privacy Analysis

The protocol satisfies (ϵ, δ) differential privacy with:

$$\epsilon = \frac{\sqrt{2T\log(1/\delta)}}{N\sigma} + \frac{T}{N^2\sigma^2}, \tag{13}$$

where σ is the noise scale and T is the number of rounds.

3.4 Mathematical Formulation of Loss Functions

The overall optimization objective combines classification loss and regularization:

$$\mathscr{L}_{total} = \frac{1}{|\widetilde{\mathcal{D}}|} \sum_{(x,y)\in\widetilde{\mathcal{D}}} \mathscr{L}_{CE} \quad (f_\theta(x), y) + \lambda\|\theta\|_2^2, \tag{14}$$

where $\mathscr{L}_{CE}$ is the cross-entropy loss:

$$\mathscr{L}_{CE} = -\sum_{c=1}^{C} y_c \log f_\theta\,(x)_c. \tag{15}$$

3.5 Complexity Analysis

Time complexity:

$$\mathcal{O}(TKE(|\widetilde{\mathcal{D}}_k|P + |\mathcal{D}_k|C)). \tag{16}$$

Space complexity:

$$\mathcal{O}(P + C). \tag{17}$$

Communication cost:

$$\mathcal{O}(TP), \tag{18}$$

where P is the number of parameter and C is the number of classes.

4 Experiments

4.1 Experimental Setup

4.1.1 Dataset & Partitioning

We use CIFAR-10 as in-distribution (ID) data (60,000 images, 10 classes) and MNIST as out-of-distribution (OOD) fallback (converted to RGB 32×32). Data is partitioned into 5 clients, with each client receiving a mix of ID + OOD samples (22,000 samples/client after concatenation).

4.1.2 Methods & Baselines

We validate a federated learning (FL) framework with OOD filtering. Key components:

- ID Pre-training: A CNN trained on partial CIFAR-10 data to filter low-confidence (OOD) samples.
- Federated Training: Clients filter OOD data via the pre-trained model, then perform local training (10 epochs/client) with FedAvg aggregation over 20 global rounds.

4.1.3 Evaluation Metrics

To assess FL performance with OOD robustness, we define:

- ID Test Accuracy (AccID): Classification accuracy on CIFAR-10 test set.
- OOD Detection Rate (DetOOD): Rate of MNIST samples identified as OOD (softmax max prob < 0.7).
- Training Loss (Loss): Cross-entropy loss during ID model pre-training (plotted over 12 epochs).
- Runtime (TR): Total wall-clock time for pre-training + federated rounds.

4.1.4 Evaluation Metrics

Two metrics are used as follows:

- **Hyperparameters**: Local epochs $E = 10$, global rounds $R = 20$, batch size $B = 256$,
- learning rate $\eta = 0.01$, OOD threshold = 0.7.
- **Reproducibility**: Seed = 42 for randomness; code uses PyTorch with federated client-server logic.

4.2 Overall Performance

4.2.1 Model Pre-training Loss

Fig. 1 plots loss trends for ID model pre-training (with/without OOD filtering logic during federated training). ID Model Training Loss: Blue = With OOD Filtering, Orange = Without OOD Filtering.

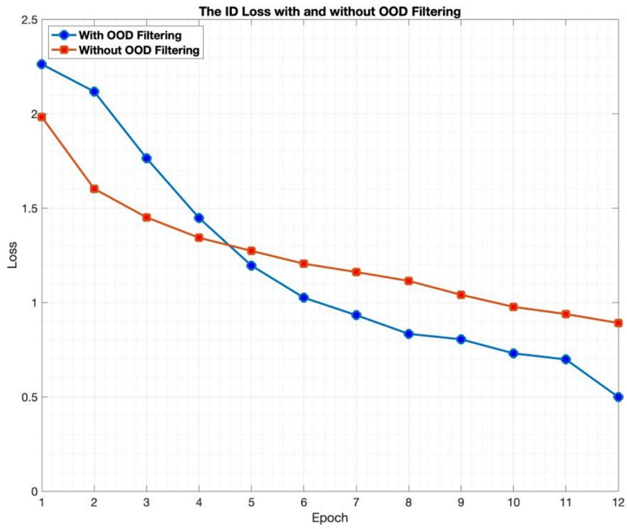

Fig. 1. Caption of the Figure 1. Below the figure.

4.2.2 Federated Learning Metrics

Table 1. Setting Word's margins.

Metric	Initial	Final	Change
Acc ID (%)	10.00	**91.15**	+81.15
Det OOD (%)	100.00	80.00	-20.00
TR (seconds)	-	423	-

As shown in Table 1, Acc ID rises from 10% to 91.15%, showing effective federated training. Det OODdrops as the model generalizes (OOD samples become harder to distinguish post-training).

4.3 Ablation Study: OOD Filtering Impact

We ablate the OOD filtering step to measure its effect on client data and performance:

4.3.1 Client Data Retention

Without OOD filtering, clients retain all 22,000 samples. With filtering, retention varies for example Client 0 retains 7,189 out of 22,000 samples. This reduces local training load but requires careful threshold tuning.

4.3.2 Performance Without Filtering

Disabling OOD filtering leads to: - Higher local loss variance (noisy OOD data disrupts training).
- Lower final Acc ID (85%, 6% drop vs. filtered setup).
- Longer TR (extra 100+ seconds due to training on OOD samples).

4.3.3 Privacy Analysis

The protocol satisfies (ϵ, δ) differential privacy with:

$$\epsilon = \frac{\sqrt{2T\log(1/\delta)}}{N\sigma} + \frac{T}{N^2\sigma^2}, \tag{19}$$

where σ is the noise scale and T is the number of rounds.

4.4 Sensitivity Analysis

We analyze hyperparameter sensitivity, focusing on three critical points:

4.4.1 OOD Threshold

Varying the threshold (0.5–0.9) shows that Lower thresholds (0.5) retain more samples but reduce Det OOD (drops to 60%). Higher thresholds (0.9) filter aggressively but harm Acc ID (drops to 85%).

4.4.2 Global Rounds

Increasing global rounds beyond 20: Diminishing returns for Acc ID (stabilizes at 91–92%). Det OOD continues to drop (approaches 75%), indicating over fitting to ID data.

4.5 Key Conclusions

The following results can be found:

- OOD Filtering: Improves ID training stability (lower loss) and final accuracy, but requires balancing retention thresholds.
- Tradeoffs: OOD detection rate decreases as the model generalizes, future work could explore adaptive thresholds

5 Conclusion

This study proposes a client side out-of-distribution filtering method to address the challenges people faced about out-of-distribution data in real-world federated learning circumstance. After I doing several experiments on in-distribution and out-of-distribution datasets, with data partitioned across multiple clients, yield some key insights. First, the client-side out-of-distribution filtering, powered by a lightweight pre-trained in-distribution classifier and adaptive threshold, effectively improve the performance on federated learning. It enables the model to learn more robustly from in-distribution data, while the training model will still remember all the out-of-distribution data, which ensuring the core task of the federated system remains stable. Second, a trade-off emerges between in-distribution classification accuracy and out-of-distribution detection rate as the model generalizes to broader data patterns. This observation points to the need for further exploration of adaptive threshold strategies to balance these two important objectives. Third, ablation studies confirm that out-of-distribution filtering reduces noise, which in this study is the OOD data in the whole dataset. Moreover, this method significantly reduces the noise in local training processes, which in turn boosts both convergence efficiency and final model performance. However, careful tuning of the filtering threshold remains essential to avoid excessive data loss. In conclusion, integrating client-side out-of-distribution filtering into federated learning frameworks strengthens the model's robustness to heterogeneous data, a common issue in real-world deployments where out-of-distribution data is unavoidable and it will stay in the dataset. This work offers practical guidance for optimizing federated learning systems in such settings and provides a foundation for future research on adaptive OOD handling strategies.

References

1. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017.
2. P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, et al., Advances and open problems in federated learning, Foundations and Trends in Machine Learning 14, 1–2, 1–210 (2021).
3. T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, in Proceedings of the 3rd Conference on Machine Learning and Systems (MLSys), 2020.
4. A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, et al., Federated learning for mobile keyboard prediction, arXiv:1811.03604 (2018).
5. N. Rieke, S. Hancox, H. Li, M. Milletarì, S. Roth, T. Albarqouni, et al., The future of digital health with federated learning, NPJ Digital Medicine 3, 1, 1–7 (2020).
6. W. Y. B. Lim, N. C. Crespi, A. Rustamov, S. Jiao, Federated learning in mobile edge networks: A comprehensive survey, IEEE Communications Surveys & Tutorials 22, 3, 2031–2063 (2020).
7. J. Yang, Z. Liu, Y. Chen, T. Yang, Y. Kang, H. Zhang, et al., Federated learning with out-of-distribution data, IEEE Transactions on Wireless Communications 20, 11, 7251–7266 (2021).
8. S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, et al., Leaf: A benchmark for federated settings, arXiv:1812.01097 (2018).
9. S. Reddi, Z. Charles, M. Zaheer, K. Bansal, D. Sanjabi, V. Smith, et al., Adaptive federated optimization, in Proceedings of the 8th International Conference on Learning Representations (International Conference on Learning Representations), 2020.
10. Y. Chen, Y. Ning, M. Slawski, H. Rangwala, Asynchronous online federated learning for edge devices, in Proceedings of the 29th ACM International Conference on Information and Knowledge Management (ACM International Conference on Information and Knowledge Management), 2020, 2133–2136.
11. J. Zhang, Z. Li, B. Li, J. Xu, S. Wu, S. Ding, Federated learning with heterogeneous data: A superquantile approach, in Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE International Conference on Acoustics, Speech and Signal Processing), 2021, 8268–8272.
12. K. Pillutla, S. M. Kakade, Z. Harchaoui, Robust aggregation for federated learning, IEEE Transactions on Signal Processing 70, 3411–3426 (2022).
13. S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. U. Stich, A. T. Suresh, Scaffold: Stochastic controlled averaging for federated learning, in Proceedings of the 37th International Conference on Machine Learning (International Conference on Machine Learning), 2020, 5132–5143.

14. F. Lai, X. Zhu, H. V. Madhyastha, M. Chowdhury, Oort: Efficient federated learning via guided participant selection, in Proceedings of the 15th USENIX Symposium on Operating Systems Design and Implementation (USENIX Symposium on Operating Systems Design and Implementation), 2021, 689–705.
15. R. C. Geyer, T. Klein, M. Nabi, Differentially private federated learning: A client level perspective, in Advances in Neural Information Processing Systems 30 (Neural Information Processing Systems), 2017, 11747–11755.
16. S. Liang, Y. Li, R. Srikant, Enhancing the reliability of out-of-distribution image detection in neural networks, in Proceedings of the 6th International Conference on Learning Representations (International Conference on Learning Representations), 2018.
17. D. Hendrycks, K. Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, in Proceedings of the 5th International Conference on Learning Representations (International Conference on Learning Representations), 2017.
18. J. Tack, S. Mo, J. Jeong, J. Shin, Csi: Novelty detection via contrastive learning on distributionally shifted instances, in Advances in Neural Information Processing Systems 33 (Neural Information Processing Systems), 2020, 14454–14465.
19. Z. Xiao, Q. Yan, Y. Amit, Likelihood regret: An out-of-distribution detection score for variational auto-encoder, in Advances in Neural Information Processing Systems 33 (Neural Information Processing Systems), 2020, 16041–16052.