FEDAUDIO: A FEDERATED LEARNING BENCHMARK FOR AUDIO TASKS

Tuo Zhang[1*], *Tiantian Feng*[1*], *Samiul Alam*[2,4],
Sunwoo Lee[3], *Mi Zhang*[4,2], *Shrikanth S. Narayanan*[1], *Salman Avestimehr*[1]

[1] Department of Electrical and Computer Engineering, University of Southern California
[2] Department of Electrical and Computer Engineering, Michigan State University
[3] Department of Computer Engineering, Inha University
[4] Department of Computer Science and Engineering, The Ohio State University

ABSTRACT

Federated learning (FL) has gained substantial attention in recent years due to data privacy concerns related to the pervasiveness of consumer devices that continuously collect data from users. While a number of FL benchmarks have been developed to facilitate FL research, none of them include audio data and audio-related tasks. In this paper, we fill this critical gap by introducing a new FL benchmark for audio tasks which we refer to as `FedAudio`. `FedAudio` includes four representative and commonly used audio datasets from three important audio tasks that are well aligned with FL use cases. In particular, a unique contribution of `FedAudio` is the introduction of data noises and label errors to the datasets to emulate challenges when deploying FL systems in real-world settings. `FedAudio` also includes the benchmark results of the datasets and a PyTorch library with the objective of facilitating researchers to fairly compare their algorithms. We hope `FedAudio` could act as a catalyst to inspire new FL research for audio tasks and thus benefit the acoustic and speech research community. The datasets and benchmark results can be accessed at `https://github.com/zhang-tuo-pdf/FedAudio`.

***Index Terms*—** federated learning, audio benchmarks

1. INTRODUCTION

Data privacy has become one of the most critical issues when dealing with personal data. Particularly, audio data, which are now widely accessible in consumer applications like Amazon Alexa, Google Assistant, and Apple Siri, can reveal a significant amount of private information about an individual.

Recently, federated learning (FL) has emerged as a privacy-preserving solution to address this pressing concern [1, 2, 3]. To facilitate FL research, as summarized in Table 1, a number of FL benchmarks have been developed in the past few years. For example, LEAF [4] is a FL benchmark that includes five datasets for tasks such as natural language processing (NLP) and computer vision (CV). TensorFlow Federated [5] expands LEAF by adding three additional datasets from NLP and CV tasks. FedML [6] introduces several benchmarks targeting Graph Neural Networks (GNN), NLP, and CV related applications. Lastly, Flamby [7] incorporates seven datasets in the healthcare domain, including both tabular and image data. Unfortunately, although these existing benchmarks have made significant contributions on facilitating FL research, none of these benchmarks include audio data and audio-related tasks.

*The first two authors contributed equally.

Table 1. Comparing FedAudio with existing FL benchmarks.

	Data Type	Noisy Data	Noisy Label
LEAF [4]	Image, Text	✗	✗
TensorFlow FL [5]	Image, Text	✗	✗
FedML [6]	Graph, Image, Text	✗	✗
Flamby [7]	Image, Tabular	✗	✗
FedAudio	**Audio**	✓	✓

In this work, we introduce `FedAudio`, a federated learning benchmark for audio tasks. `FedAudio` includes four representative and commonly used datasets from three popular audio tasks – keyword spotting, speech emotion recognition, and sound event classification – that are well aligned with FL. Unlike existing FL benchmarks listed in Table 1 that only include datasets with clean data and accurate labels, in `FedAudio`, we introduce data noises and label errors to the datasets to emulate scenarios when deploying FL systems in real-world settings. This is a key difference and a unique contribution of `FedAudio` compared to existing FL benchmarks. In addition, `FedAudio` includes the benchmark results of the datasets and a PyTorch library to facilitate researchers to fairly compare their algorithms. We hope `FedAudio` could become the reference FL benchmark for audio tasks, and facilitate future FL research in the acoustic and speech research community.

2. MOTIVATION AND DESIGN CONSIDERATIONS

Although FL research has made significant progress over the past few years, the FL research and applications on audio-related tasks are relatively limited. Most of the FL works on audio-related tasks are conducted either on private datasets [12, 13] or using different experimental setups [14, 15, 16, 17], making it difficult for researchers to fairly compare their methods and push the frontier forward. Such gap motivates us to develop a high-quality FL benchmark for audio tasks to accelerate audio-based FL research.

However, the design of such benchmark requires some unique considerations. First, the selected datasets need to be both representative and diversified in dimensions such as data size and number of classes, and the selection of the audio tasks needs to align with the use cases of FL. Second, the benchmark needs to be designed to replicate real-world situations as much as possible. As such, the FL algorithms to be developed on top of the benchmark can be more relevant for deployment in real-world settings.

Table 2. Overview of the datasets included in FedAudio.

Dataset	Task	Pre-processing Method	Non-IID Partition Scheme	# clients	# samples	# labels
Google Commands [8]	Keyword Spotting	Mel Spectrogram	Speaker ID	2,618	105,829	35
IEMOCAP [9]	Speech Emotion Recognition	Pretrained APC model	Actor ID	10	2,943	4
CREMA-D [10]	Speech Emotion Recognition	Pretrained APC model	Speaker ID	91	4,798	4
Urban Sound [11]	Sound Event Classification	Mel Spectrogram	Dirichlet Distribution	50	8,732	10

3. FEDAUDIO DESIGN

3.1. Datasets

As listed in Table 2, `FedAudio` includes four audio datasets (Google Speech Commands, IEMOCAP, CREMA-D, Urban Sound) that target three different audio tasks (keyword spotting, emotion recognition, event classification). We select these datasets and audio tasks for two primary reasons. First, the selected audio tasks are well aligned with FL and are regarded as some of the killer applications of FL. Second, the selected datasets are among the most representative and commonly used datasets for the corresponding tasks. Moreover, the selected datasets cover both small and large numbers of clients, data samples, and class labels, and therefore contribute to a diversified and comprehensive collection. It should be noted that we do not include the task of automatic speech recognition (ASR). This is mainly because the task of ASR can not be well supported under FL due to its demands on computational resources for large model training and end users for large-amount data labeling [18]. In the following, we describe each dataset along with its pre-possessing method and non-IID data partitioning scheme.

3.1.1. *Google Speech Commands*

The Google Speech Commands dataset [8] is designed for developing basic voice interfaces for the task of keyword spotting. The dataset includes 35 common words from the everyday vocabulary such as ”Yes”, ”No”, ”Up”, and ”Down”. It contains a total of 105,829 audio recordings collected from 2,618 speakers. The training set includes the recordings from 2,112 speakers and the test set includes the recordings from the rest. To pre-process the raw audio data, a sequence of overlapping Hamming windows is applied to the raw speech signal with a time shift of 10 ms. We calculate the discrete Fourier transform (DFT) with a frame length of 1,024 and compute the Mel-spectrogram with a dimension of 128. The Mel-spectrogram is used for training the keyword spotting model. The Google Speech Commands dataset is partitioned over speaker IDs, making the dataset naturally non-IID distributed.

3.1.2. *IEMOCAP*

The IEMOCAP dataset [9] is a multimodal dataset designed for the task of emotion recognition. The dataset contains video, speech, and motion capture of face of emotional expressions collected from ten actors (five males, five females). We extracted the audio component of the IEMOCAP dataset and followed [19, 16] to focus on the more challenging improvised sessions and four most frequently occurring emotion labels (neutral, sad, happiness, and anger) for the task of speech emotion recognition (improvised). We followed [16] and used the pre-trained autoregressive predictive coding (APC) features [20] as the input for training the speech emotion recognition (SER) model. The dataset is partitioned by the actor ID.

3.1.3. *CREMA-D*

Similar to IEMOCAP, the CREMA-D dataset [10] is also a multimodal dataset for emotion recognition. The corpus includes audio and visual data of utterances spoken under five emotional expressions: happy, sad, anger, fear, and neutral. Similar to the previous work in [16], we choose neutral, sad, happiness, and anger emotions as the candidate classification labels. Compared to IEMOCAP, CREMA-D includes many more actors (91 actors, 48 of whom are male, and 43 are female), and thus can emulate a large-scale client pool in FL. Again, we extracted the audio component of the dataset, used the pre-trained APC features for training the SER models, and partitioned the dataset by actor ID.

3.1.4. *Urban Sound*

Urban-Sound [11] is an audio database with 8,732 labeled sound recordings from ten urban sound classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. The audio recordings are spitted into ten folds for training and testing. We further divide each recording into 3-second segments, with an overlap of 0.5 second (in training) and 1 second (in testing) between segments. We extract the Mel-spectrograms using the same setting as described in Section 3.1.1. To create non-IID data distributions, we partitioned the dataset into 50 subsets using Dirichlet distribution with $\alpha \in \{0.1, 0.5\}$ to control the level of non-IID where $\alpha = 0.1$ corresponds to high data heterogeneity and $\alpha = 0.5$ corresponds to low data heterogeneity.

3.2. Data Noises and Label Errors

When deploying FL systems in real-world scenarios, the collected audio data may be subject to interference from ambient noises. Moreover, end users frequently make errors in labeling the audio data, especially for challenging tasks such as emotion recognition. Unlike existing FL benchmarks listed in Table 1 that only include datasets with clean data and accurate labels, in `FedAudio`, we introduce data noises and erroneous labels to the datasets to emulate real-world scenarios. This is a key difference and a unique contribution compared to existing FL benchmarks.

3.2.1. *Data Noises*

Background noises are everywhere in real-world environments. To study the impact of data noises on the performance of FL, we add additive white Gaussian noises (AWGN) to the audio data of the four datasets described in Section 3.1. Specifically, we followed [21] to use the signal-to-noise ratio (SNR) as a measurement for data noise level, which is defined as the ratio of the power of the original signal to the power of background noise, and expressed in decibel (dB). We would also highlight that FedAudio allows emulating non-stationary noises like environmental noises, but due to the limited space in the paper, we decided to focus on the analysis of data noises that belongs to the Gaussian noises.

3.2.2. *Label Errors*

Label error refers to the mismatch between the ground truth label of a data sample and the label provided by the end user. To emulate label error, we augment the ground truth labels of a dataset with a transition matrix Q, where $Q_{ij} = P(\hat{y} = j \mid y = i)$ denotes the probability that the ground truth label i is changed to a different label j. To do so, we followed [22] to control the generation of the transition matrix Q using two parameters: error ratio and error sparsity. Specifically, the error ratio quantifies the total amount of noise in a dataset, and is defined as the sum of the probabilities that the ground truth label i flips to the other labels. An error ratio of 0 represents a dataset with all the data accurately labeled, whereas an error ratio of 1 implies that all the data in a dataset have wrong labels. On the other hand, error sparsity quantifies the distribution of the label errors, which is defined as the fraction of zeros in the off-diagonal of the transition matrix Q. An error sparsity of 0 represents each element in Q is non-zero, which implies one class could be altered to any other class. An error sparsity of 1 indicates that there is no label error as all off-diagonals are equal to zero.

3.3. Library

We have created a library to facilitate the use of `FedAudio` benchmarks. As illustrated in Figure 1, our library incorporates a *FL Feature Manager* that adds data and label noise to emulate real-world scenarios. Besides AWGN, we also include ESC-50 dataset [23] in the data noise part as the non-stationary background noise data. The *Pre-processing Manager* supports not only conventional frequency-based features (e.g., Mel-frequency cepstral coefficients (MFCC), Mel Spectrograms) and knowledge-based speech features (e.g., pitch) but also deep audio representations from pre-trained models. The *Data Spliter* is responsible of partitioning the dataset into non-IID distribution via either natural partitioning (e.g., user ID) or manual partitioning (e.g., Dirichlet distribution). Finally, our library supports a handful of commonly used FL optimizers such as FedAvg [24] and FedOPT [25], and allows users to import new models to be trained on the included datasets. We include FedML open source software [6] (`https://github.com/FedML-AI/FedML`) as the FL framework to implement the FedAudio. The source codes and user guides of FedAudio are available at `https://github.com/zhang-tuo-pdf/FedAudio`.

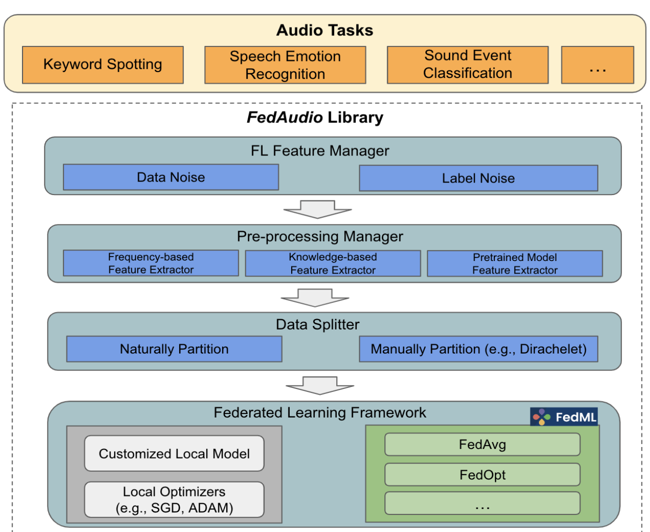

Fig. 1. Structure of FedAudio library.

Table 3. Benchmarks on clean data. We denote the average and the standard deviation of the performance by the notation: avg. (std.)%. # Samp Cli stands for the number of sampled clients per round.

Dataset	# Samp Cli	Centralized	FedAvg	FedOPT
Google Command	—	91.36 (0.30)%	—	—
	106 (5%)	—	88.62 (0.51)%	91.05 (0.23)%
	212 (10%)	—	88.44 (0.41)%	90.82 (0.09)%
	424 (20%)	—	89.00 (0.24)%	90.91 (0.29)%
IEMOCAP	—	58.95 (4.95)%	—	—
	8 (100%)	—	53.80 (6.61)%	55.45 (6.78)%
CREMA-D	—	70.05 (1.12)%	—	—
	7 (10%)	—	68.80 (2.58)%	64.78 (2.94)%
	21 (30%)	—	68.93 (3.34)%	66.51 (2.30)%
	36 (50%)	—	67.63 (3.87)%	64.93 (4.33)%
Urban Sound ($\alpha = 0.5$)	—	62.83 (4.17)%	—	—
	10 (20%)	—	59.23 (4.21)%	57.53 (5.46)%
	25 (50%)	—	59.52 (4.84)%	61.16 (7.72)%
Urban Sound ($\alpha = 0.1$)	—	62.83 (4.17)%	—	—
	10 (20%)	—	52.49 (7.34)%	51.13 (5.46)%
	25 (50%)	—	54.01 (6.87)%	56.07 (7.84)%

4. EXPERIMENTS

In this section, we provide benchmark results of the datasets listed in Table 2. To ensure the consistency of the results, we used the same model for all four datasets. Specifically, the model consists of two convolution layers followed by one Gated Recurrent Units (GRU) layer. An average pooling layer is connected to the GRU output, which is then fed through two dense layers to generate the predictions. The detailed hyperparameter settings are listed in the GitHub link. For Google Speech Command, we report the mean and std. of the accuracy over five trials with different seeds. For the other three datasets, we evaluate each of the train/test splits provided by the datasets and report the mean and std. of F1 score.

4.1. Benchmarks on Clean Audio Data

First, we provide benchmark results on clean audio data. Specifically, we benchmark the performance under different client sample ratios (except the IEMOCAP dataset due to limited number of speakers) and two FL optimizers (FedAvg and FedOPT), and compare it against centralized training. For fair comparison, we use the same local model and optimizer configuration for both centralized and federated training. We aim to benchmark the following performance: (1) What is the performance gap between centralized and federated settings? (2) What is the impact of client sample ratio on the training performance? (3) What is the impact of FL optimizers on the training performance? (4) How does the level of data heterogeneity affect the training performance?

Benchmark Results: Table 3 summarizes our results. (1) The gaps between centralized and FL settings across all datasets are between 0.31% to 11.7%. For the same speech emotion recognition task, IEMOCAP achieves lower accuracy than CREMA-D. One plausible reason is that data included in IEMOCAP are from actors' improvisation. In contrast, the data from CREMA-D are strictly based on the script reading, which provides less variation compared to IEMOCAP. (2) The increasing of client sample ratio does not necessarily bring performance gains. (3) Both FL optimizers achieve competitive performance. (4) There is a gap in training performance between high and low data heterogeneity and higher data heterogeneity has a negative effect on the training performance.

Table 4. Benchmarks on data with noises. ">" means the target accuracy was not reached.

Dataset	SNR	ΔMetric	Target Acc.	Training Rounds
Google Command	10dB	$\downarrow$27.12 (2.11)%	65%	> 5000
			75%	> 5000
	20dB	$\downarrow$12.74 (0.64)%	65%	910 (1.52×)
			75%	2110 (2.32×)
	30dB	$\downarrow$4.48 (0.39)%	65%	775 (1.29×)
			75%	1485 (1.63×)
IEMOCAP	10dB	$\downarrow$5.99 (1.90)%	35%	66 (1.32×)
			45%	142 (1.65×)
	20dB	$\downarrow$5.56 (2.32)%	35%	59 (1.18×)
			45%	111 (1.29×)
	30dB	$\downarrow$4.82 (2.78)%	35%	55 (1.10×)
			45%	103 (1.20×)
CREMA-D	10dB	$\downarrow$9.11 (2.24)%	55%	58 (1.45×)
			65%	> 200
	20dB	$\downarrow$4.43 (2.40)%	55%	47 (1.18×)
			65%	149 (1.73×)
	30dB	$\downarrow$1.58 (2.40)%	55%	42 (1.05×)
			65%	128 (1.48×)
Urban Sound ($\alpha = 0.1$)	10dB	$\downarrow$28.07 (3.43)%	20%	184 (3.93×)
			25%	> 300
	20dB	$\downarrow$25.51 (4.23)%	20%	145 (3.09×)
			25%	211 (3.44×)
	30dB	$\downarrow$24.59 (4.43)%	20%	157 (3.56×)
			25%	176 (2.87×)

Table 5. Benchmarks on data with label errors. ">" means the target accuracy was not reached. **E.R** stands for error ratio.

Dataset	E.R.	ΔMetric	Target Acc.	Training Rounds
Google Command	0.1	$\downarrow$0.38 (0.16)%	75%	1050 (1.75×)
			85%	2450 (2.69×)
	0.3	$\downarrow$2.23 (0.25)%	75%	1460 (2.43×)
			85%	3710 (4.08×)
	0.5	$\downarrow$5.25 (0.28)%	75%	2335 (3.89×)
			85%	> 5000
IEMOCAP	0.1	$\downarrow$4.38 (4.10)%	35%	56 (1.13×)
			45%	106 (1.24×)
	0.3	$\downarrow$9.64 (1.91)%	35%	80 (1.61×)
			45%	> 200
	0.5	$\downarrow$19.43 (6.53)%	35%	> 200
			45%	> 200
CREMA-D	0.1	$\downarrow$0.39 (2.85)%	55%	51 (1.28×)
			65%	112 (1.30×)
	0.3	$\downarrow$4.68 (1.16)%	55%	84 (2.00×)
			65%	> 200
	0.5	$\downarrow$8.68 (3.76)%	55%	134 (3.36×)
			65%	> 200
Urban Sound ($\alpha = 0.1$)	0.1	$\downarrow$0.71 (7.12)%	35%	101 (1.21×)
			45%	186 (1.32×)
	0.3	$\downarrow$4.62 (7.49)%	35%	153 (1.83×)
			45%	211 (1.50×)
	0.5	$\downarrow$10.98 (7.41)%	35%	221 (2.66×)
			45%	> 300

4.2. Benchmarks on Noisy Audio Data

Next, we provide benchmark results on noisy audio data. Besides model accuracy/F1 score, we also use the round-to-accuracy metric, which is defined as the communication rounds needed for training the model to reach a target accuracy. As shown in Section 4.1, we observed that both client sampling rate and FL optimizer have limited impact on benchmarking performance. Therefore, we use the smallest client sample ratio combined with the FedAvg optimizer from Table 3 as the baseline. For Urban Sound, we consider the more challenging high data heterogeneity setting. ΔMetric denotes the model accuracy/F1 score difference between the baseline and the results on noisy audio data. We aim to benchmark the following performance: (1) What is the impact of data noises on model accuracy/F1 score and round-to-accuracy? (2) What is the impact of SNR on training performance?

Benchmark Results: Table 4 summarizes our results. (1) AWGN not only degrades model accuracy/F1 score but also increases the required training rounds to the target performance. One notable point is that the AWGN significantly degrades the model performance for Urban Sound, which reveals the vulnerability of sound event classification to the white background noise. (2) With the increment of SNR level, we find improvements in both model accuracy/F1 score and round-to-accuracy. These results indicate that noise cancellation is necessary before feeding the audio data to the FL algorithm. However, even with 30dB SNR, although the model performance decreases by a small margin, the convergence speed is delayed, which brings attention to developing more robust FL strategies on the system side under these settings.

4.3. Benchmarks on Audio Data with Label Errors

Finally, we provide benchmark results on audio data with label errors. For each dataset, we set the error ratio in the range of 0.1 to 0.5 with a step size of 0.2, and set the error sparsity to 0.4 to emulate non-IID distribution. Again, for Urban Sound, we consider the high data heterogeneity setting. We aim to benchmark the impact of label errors on model accuracy/F1 score as well as round-to-accuracy.

Benchmark Results: Table 5 summarizes our results. Among all the datasets, with the error ratio below 0.3, the model performance remains relatively high, but the learning process takes much more compared to clean data. When the error ratio reaches 0.5, the model performance drops substantially. The results imply that as the amount of correctly labeled data samples is above a specific threshold, the model could still be trained robustly but with slow convergence speed and a slight decrease in the model performance. We can also find that each dataset has a different performance drop at an error ratio of 0.5. For example, the final model performance drop is 5.19% in Google Command but is 19.43% in IEMOCAP. Since the baseline Google Speech Command performance is much higher than IEMOCAP, we hypothesize that this difference is introduced by the task characteristics, where the model performance of a relatively more challenging speech task is less prone to label noises.

5. CONCLUSION

We introduced a federated learning benchmark for audio tasks named `FedAudio`. Currently, `FedAudio` includes four representative datasets of three important audio tasks. Besides the clean audio data, our paper investigates how real-world challenges, including data noises and label errors, affect the FL training performance. We hope `FedAudio` could act as a catalyst to inspire new FL research in the acoustic and speech research community.

6. ACKNOWLEDGEMENT

This work is in part supported by research gifts from Intel and Qualcomm, Meta Faculty Research Award, USC Amazon Center for Secure and Trusted AI, and INHA UNIVERSITY Research Grant.

7. REFERENCES

[1] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al., "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[2] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al., "A field guide to federated optimization," *arXiv preprint arXiv:2107.06917*, 2021.

[3] Tuo Zhang, Lei Gao, Chaoyang He, Mi Zhang, Bhaskar Krishnamachari, and Salman Avestimehr, "Federated learning for the internet of things: Applications, challenges, and opportunities," *IEEE Internet of Things Magazine*, vol. 5, pp. 24–29, 2022.

[4] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konecný, H. B. McMahan, Virginia Smith, and Ameet S. Talwalkar, "Leaf: A benchmark for federated settings," *ArXiv*, vol. abs/1812.01097, 2018.

[5] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: a system for large-scale machine learning.," in *Osdi*. Savannah, GA, USA, 2016, vol. 16, pp. 265–283.

[6] Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr, "Fedml: A research library and benchmark for federated machine learning," *arXiv preprint arXiv:2007.13518*, 2020.

[7] Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, et al., "Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[8] Pete Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *ArXiv*, vol. abs/1804.03209, 2018.

[9] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Ebrahim (Abe) Kazemzadeh, Emily Mower Provost, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan, "Iemocap: interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, vol. 42, pp. 335–359, 2008.

[10] Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma, "Crema-d: Crowd-sourced emotional multimodal actors dataset," *IEEE transactions on affective computing*, vol. 5, no. 4, pp. 377–390, 2014.

[11] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM International Conference on Multimedia (ACM-MM'14)*, Orlando, FL, USA, Nov. 2014, pp. 1041–1044.

[12] Meng Feng, Chieh-Chi Kao, Qingming Tang, Ming Sun, Viktor Rozgic, Spyros Matsoukas, and Chao Wang, "Federated self-supervised learning for acoustic event classification," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 481–485, 2022.

[13] Xiaodong Cui, Songtao Lu, and Brian Kingsbury, "Federated acoustic modeling for automatic speech recognition," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6748–6752, 2021.

[14] Dhruv Guliani, Françoise Beaufays, and Giovanni Motta, "Training speech recognition models with federated learning: A quality/cost framework," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3080–3084, 2021.

[15] Dimitrios Dimitriadis, Ken'ichi Kumatani, Robert Gmyr, Yashesh Gaur, and Sefik Emre Eskimez, "A federated approach in training acoustic models," in *INTERSPEECH*, 2020.

[16] Tiantian Feng and Shrikanth S. Narayanan, "Semi-fedser: Semi-supervised learning for speech emotion recognition on federated learning using multiview pseudo-labeling," *ArXiv*, vol. abs/2203.08810, 2022.

[17] Yan Gao, Titouan Parcollet, Javier Fernández-Marqués, Pedro Porto Buarque de Gusmão, Daniel J. Beutel, and Nicholas D. Lane, "End-to-end speech recognition from federated acoustic models," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7227–7231, 2022.

[18] Yan Gao, Javier Fernández-Marqués, Titouan Parcollet, Abhinav Mehrotra, and Nicholas D. Lane, "Federated self-supervised speech representations: Are we there yet?," in *INTERSPEECH*, 2022.

[19] Yuanyuan Zhang, Jun Du, Zirui Wang, and Jian shu Zhang, "Attention based fully convolutional network for speech emotion recognition," *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1771–1775, 2018.

[20] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James R. Glass, "An unsupervised autoregressive model for speech representation learning," in *INTERSPEECH*, 2019.

[21] Chloé Clavel, Thibaut Ehrette, and Gaël Richard, "Events detection for an audio-based surveillance system," in *2005 IEEE International Conference on Multimedia and Expo*. IEEE, 2005, pp. 1306–1309.

[22] Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *J. Artif. Intell. Res.*, vol. 70, pp. 1373–1411, 2021.

[23] Karol J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. pp. 1015–1018, ACM Press.

[24] H. B. McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.

[25] Sashank J. Reddi, Zachary B. Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konecný, Sanjiv Kumar, and H. B. McMahan, "Adaptive federated optimization," *ArXiv*, vol. abs/2003.00295, 2021.

8. APPENDIX

8.1. Hyperparameter Settings

We searched for the client learning rate in a range from 10^{-6} to 10^{0}, the server learning rate in a range from 10^{-4} to 10^{0}, and input batch size in a range from 5 to 30, and total training round in a range from 50 to 5000. After hyperparameter searching, we fixed the batch size for all datasets to 16 and the local epoch to 1 for all experiments.

8.1.1. Training with clean signals

When we train the model with clean signals, we select both FedAvg and FedOpt as the server aggregator functions. For FedOpt, we select the ADAM as the server optimizer. The other hyperparameters are shown in Table 6.

Table 6. Hyperparameters for training FL models with clean audio signals.

Dataset	Agg.	Sample rate	lr	Server lr	Round
Google Command	FedAvg	5%	0.1	-	5000
		10%	0.3	-	5000
		20%	0.2	-	5000
	FedOpt	5%	0.05	0.001	5000
		10%	0.01	0.001	5000
		20%	0.01	0.001	5000
IEMOCAP	FedAvg	100%	0.01	-	200
	FedOpt	100%	0.01	0.001	50
Crema-D	FedAvg	10%	0.1	-	200
		30%	0.1	-	200
		50%	0.1	-	200
	FedOpt	10%	0.1	0.001	200
		30%	0.1	0.001	200
		50%	0.1	0.001	200
Urban Sound	FedAvg	20%	0.075	-	300
		50%	0.075	-	300
	FedOpt	20%	0.1	0.001	300
		50%	0.1	0.001	300

8.1.2. Training with SNR noisy signals

When we train the model with SNR noisy signals, we select FedAvg as the server aggregator function. For Google command related experiments, the client learning rate is 0.1 and the communication round is 5000. For IEMOCAP related experiments, the client learning rate is 0.01 and the communication round is 200. For CREMA-D related experiments, the client learning rate is 0.1 and the communication round is 200. Finally, for Urban Sound, the client learning rate is 0.075 and the communication round is 300.

8.1.3. Training with label noisy signals

In this section, we select FedAvg as the server aggregator function. For Google command related experiments, the client learning rate is 0.1 and the communication round is 5000. For IEMOCAP related experiments, the client learning rate is 0.01 and the communication round is 200. For CREMA-D related experiments, the client learning rate is 0.1 and the communication round is 200. Finally, for Urban Sound, the client learning rate is 0.075, and the communication round is 300.