INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Zero-Shot Federated Learning with New Classes for Audio Classification

Gautham Krishna Gudur[1], *Satheesh Kumar Perepu*[2]

[1]Global AI Accelerator, Ericsson, Chennai, India
[2]Ericsson Research, Chennai, India

gautham.krishna.gudur@ericsson.com, perepu.satheesh.kumar@ericsson.com

Abstract

Federated learning is an effective way of extracting insights from different user devices while preserving the privacy of users. However, new classes with completely unseen data distributions can stream across any device in a federated learning setting, whose data cannot be accessed by the global server or other users. To this end, we propose a unified zero-shot framework to handle these aforementioned challenges during federated learning. We simulate two scenarios here – 1) when the new class labels are not reported by the user, the traditional FL setting is used; 2) when new class labels are reported by the user, we synthesize *Anonymized Data Impressions* by calculating class similarity matrices corresponding to each device's new classes followed by unsupervised clustering to distinguish between new classes across different users. Moreover, our proposed framework can also handle statistical heterogeneities in both labels and models across the participating users. We empirically evaluate our framework on-device across different communication rounds (FL iterations) with new classes in both local and global updates, along with heterogeneous labels and models, on two widely used audio classification applications – keyword spotting and urban sound classification, and observe an average deterministic accuracy increase of ∼4.041% and ∼4.258% respectively.

Index Terms: keyword spotting, urban sound classification, federated learning, new class identification, zero-shot learning, on-device learning

1. Introduction

Deep learning for audio classification is a broad research area with applications like Keyword Spotting (KWS), urban sound identification, etc. KWS is an important application for detecting keywords of importance to specific users, which could be used as voice commands to on-device personal assistants such as Amazon's Alexa, Apple's Siri, etc. [1]. Urban environment sound classification is another interesting application particularly in context-aware computing, urban informatics [2]. The emergence of deep neural networks have conveniently alleviated problems of creating shallow (hand-picked) features to achieve state-of-the-art performance in such acoustic classification tasks [3, 4]. With the recent compute capabilities vested in resource-constrained devices, there is a huge research focus on audio classification using on-device deep learning [5, 6].

Such applications require characterization of insights across numerous user devices for personalization, and collaborative on-device deep learning becomes necessary. Federated Learning (FL) is a decentralized method of training neural networks by securely sharing model updates with a server without the need to transfer sensitive local user data [7, 8]. On-device federated learning has been an active area of research addressing challenges on secure communication protocols, optimization, privacy-preserving networks, etc. [9, 10]. However, handling new/unseen classes in local devices and training them in an FL setting for the global model to possess characteristics of the new classes is a challenging task, since data transfer from local device to server and vice versa is not feasible. Moreover, the new class information of one user is not known among the other users as well, hence the new classes could be similar or different between the users. In addition, there are multiple statistical heterogeneities like model heterogeneities (ability of end-users to architect their own local models), label heterogeneities and non-IIDness across various communication rounds/FL iterations (disparate data and label distributions across devices).

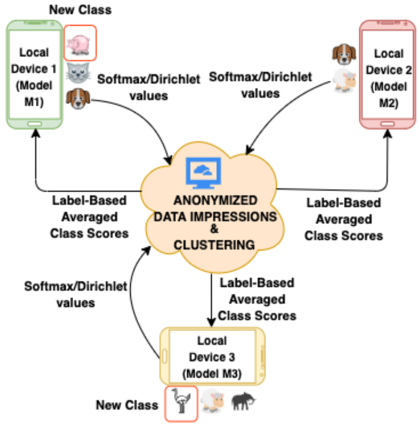

New Class
Local Device 1 (Model M1)
Softmax/Dirichlet values
Softmax/Dirichlet values
Local Device 2 (Model M2)
Label-Based Averaged Class Scores
ANONYMIZED DATA IMPRESSIONS & CLUSTERING
Label-Based Averaged Class Scores
Softmax/Dirichlet values
Label-Based Averaged Class Scores
Local Device 3 (Model M3)
New Class

Figure 1: *Architecture of the proposed Federated Learning framework with new classes streaming in across different users.*

One way of handling model heterogeneities and independence in a federated learning setting is by using knowledge distillation [11] with a common student model architecture on each local device [12]. Label and model heterogeneities are handled in an inertial Human Activity Recognition scenario in [13]. Federated learning for keyword spotting, and new class learning and identification in various speech recognition settings are addressed in [14, 15, 16]. A new augmentation technique to reduce false reject rates is proposed in [17], and it addresses algorithmic constraints in FL-KWS training to label examples with no visibility. However, the scope of our proposed work is different in the nature that it primarily addresses identification and similarity detection of new labels in a zero-shot manner when heterogeneous label and model distributions exist across various users and FL iterations. To the best of our knowledge, we are the first to discuss new label identification in FL settings with statistical heterogeneities for audio classification.

Our scientific contributions are: **(1)** A framework with zero-shot learning mechanism by synthesizing *Anonymized Data Impressions* from class similarity matrices to identify new classes for keyword spotting and urban sound detection in

http://dx.doi.org/10.21437/Interspeech.2021-2264

on-device FL settings. **(2)** Provide two scenarios for label acquisition – when class labels are reported by user, and when class labels are not, and propose unsupervised clustering to identify and differentiate between newly reported classes. **(3)** Handling statistical heterogeneities such as heterogeneous distributions in labels, data and models across devices and FL iterations.

2. Our Approach

In this section, we discuss the problem formulation of new classes in FL, and our proposed framework (Algorithm 1) to handle the same. The overall architecture is given in Figure 1.

Algorithm 1 Our Proposed Framework

```
Input: Public Dataset $\mathcal{D}_0\{x_0, y_0\}$, Private Datasets $\mathcal{D}^i_m$, Total users $M$, Total iterations $I$, LabelSet $l_m$ for each user, Overall Public LabelSet $Y$,
Output: Trained Model scores $f^I_G$
Initialize $f^0_G = \mathbf{0}$ (Global Model Scores)
for $i = 1$ to $I$ do
    for $m = 1$ to $M$ do
        Build: Model $\mathcal{D}^i_m$ and predict $f_{\mathcal{D}^i_m}(x_0)$
        Local Update:
        Choice 1: New classes are not reported
        $f_{\mathcal{D}^i_m}(x_0) = f^I_G(x_0^{l_m}) + \alpha f_{\mathcal{D}^i_m}(x_0)$, where $f^I_G(x_0^{l_m})$ are global scores of $l_m$ with $m^{th}$ user, $\alpha = \frac{len(\mathcal{D}^i_m)}{len(\mathcal{D}_0)}$
        Choice 2: New classes are reported
        Train a new model with $\mathcal{D}_0$ and $\mathcal{D}^i_m$ (new data) together, and send weights of the last layer ($\mathbf{W}^i_m$) to global user.
    end for
    Global Update:
    Choice 1: No user reports new classes
    Update label wise
    $f^{i+1}_G = \sum_{m=1}^{M} \beta_m f_{\mathcal{D}^i_m}(x_0)$, where
    $\beta = \begin{cases} 1 & \text{If labels are unique} \\ \text{acc}(f_{\mathcal{D}^{i+1}_m}(x_0)) & \text{if labels are not unique} \end{cases}$
    where $\text{acc}(f_{\mathcal{D}^{i+1}_m}(x_0))$ is the accuracy metric, defined by the ratio of correctly classified samples to total samples for a given local model.
    Choice 2: Any user reports new classes
    Create Data Impressions (DI) for each user $m$ with weights $\mathbf{W}^i_m$ (Section 2.2). Average DI of all users with new classes, $\mathbf{X}^i = \sum_{m \in M_{S_k}} \mathbf{X}^i_m$, where $M_{S_k}$ is set of users with new label $k$.
    Perform k-medoids clustering on $\mathbf{X}^i$ across $M_{S_k}$. Number of clusters = Number of new labels ($l_{new}$).
    Update public dataset with new DI ($\mathbf{X}^i$), $\mathcal{D}_{new} = \mathcal{D}_0 \bigcup \mathbf{X}^i$, add $l_{new}$ to $l_m$ and $Y$.
end for
```

2.1. Problem Formulation

We assume the following scenario in federated learning. Suppose there are M nodes (devices) in the FL network, holding private local data $\mathcal{D}_i = \{x_{i,j}, y_{i,j}\}$ where i is the FL iteration and j is the user index. Each node consists of public data $\mathcal{D}_0 = \{x_0, y_0\}$. The public data is assumed to be present across the global and all local users as discussed in [12] to handle the various statistical (model) heterogeneities which is a common phenomena in FL. The overall label-set of public dataset is $Y = \{y_0\}$, which represent its unique labels. We re-purpose this public dataset as test set and do not expose it to local models during FL training iterations, but expose only during testing for consistency. Our work's main contribution is to propose a framework to identify new labels across different users without transferring private data in FL setting. We also assume each user can stream data with new labels at any iteration which does not belong to public label-set Y, i.e. $\mathbf{y}_{i,j} \notin Y$. In other words, the global user has no idea of these new labels.

2.2. Anonymized Data Impressions

The main challenge/objective is to detect similar labels across different users in FL heterogeneous settings without the knowledge of local user data. This necessitates us to construct anonymized data without transferring raw sensitive data, and identify new class similarities on the anonymized data. We motivate our framework from the creation of Data Impressions (DI) using zero-shot learning as proposed in [18] to compute *Anonymized Data Impressions*. Let us assume a model $\mathcal{M}$ with input $\mathbf{X}$ and output $\mathbf{y}$, where $\mathbf{X} \in \mathcal{R}^{M \times N}$ is the set of features and $\mathbf{y} \in \mathcal{R}^M$. Now, the anonymized feature set $\bar{\mathbf{X}}$, which has same properties of $\mathbf{X}$, can be synthesized in two steps:

(a) Sample Softmax Values: The first step is to sample the softmax values from the Dirichlet distribution [19]. The Class Similarity Matrix (*CSM*) is created which contains important information on how similar the classes are to each other. If the classes are similar, we expect the softmax values are concentrated over these labels. *CSM* is obtained by considering the weights of the model's last layer. Typically, any classification model has the final layer as fully-connected layer with a softmax non-linearity. If the classes are similar, we find similar weights between connections of the penultimate layer to the nodes of the classes [18]. The Class Similarity Matrix is constructed as,

$$C(i,j) = \frac{\mathbf{w}_i^T \mathbf{w}_j}{||\mathbf{w}_i|| ||\mathbf{w}_j||} \tag{1}$$

where $\mathbf{w}_i$ is the vector of weights connecting the previous layer nodes to the class node i. $\mathbf{C} \in \mathcal{R}^{\mathcal{K} \times \mathcal{K}}$ is the Class Similarity Matrix for K classes. We then sample the softmax values as,

$$\text{Softmax} = Dir(K, C) \tag{2}$$

where C is concentration parameter which controls the spread of softmax values over class labels.

(b) Creating Anonymized Data Impressions: Let $\mathbf{Y}^k = [\mathbf{y}^k_1, \mathbf{y}^k_2, \cdots, \mathbf{y}^k_N] \in \mathcal{R}^{K \times N}$ be the N softmax vectors corresponding to class k, sampled from Dirichlet distribution from previous step. Once we obtain the softmax values, we compute the synthesized data features (Data Impressions) by solving the following optimization problem using model $\mathcal{M}$ and sampled softmax values $\mathbf{Y}^k$

$$\bar{\mathbf{x}} = \arg\min_{\mathbf{x}} L_{CE}(\mathbf{y}^k_i, \mathcal{M}(\mathbf{x})) \tag{3}$$

To solve this optimization problem, we initialize the input $\mathbf{x}$ to be random input and iterate until cross-entropy loss (L_{CE}) minimization. This process is repeated for all K categories. In this way, anonymized data impressions are created for each class without the visibility of original input data. We use the TensorFlow framework [20] for all our experiments.

Table 1: *Model Architectures (filters in each layer), Labels and Audio frames per FL iteration across user devices for both datasets. Note the disparate model architectures and labels across users.*

	User 1	User 2	User 3	Global User
Architecture	2-Layer CNN (16, 32) Softmax Activation	3-Layer CNN (16, 16, 32) ReLU Activation	3-Layer Depth-Separable CNN (16, 16, 32) ReLU Activation	–
Keywords	{Yes, No, Up, Down}	{Up, Down, Left, Right}	{Left, Right, On, Off}	{Yes, No, Up, Down, Left, Right, Left, Right, On, Off}
Keyword Frames per iteration	{200-300, 200-300, 200-300, 200-300}	{200-300, 200-300, 200-300, 200-300}	{200-300, 200-300, 200-300, 200-300}	{300*8} = 2400
Sounds	{air conditioner, car horn, children playing}	{children playing, dog bark, drilling }	{drilling, engine idling, gun shot, jackhammer}	{air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer}
Sound Frames per iteration	{40-50, 40-50, 40-50}	{40-50, 40-50, 40-50}	{40-50, 40-50, 40-50, 40-50}	{50*8} = 400

2.3. Proposed Framework

There are three steps in our proposed framework (Algorithm 1).

(a) Build: Each local user creates their own model with their local private data for a specific iteration.

(b) Local Update: In this step, if new classes are not reported, we perform simple weighted α-update [21], where α governs the contributions of new and old models across FL iterations shown in Algorithm 1 Choice 1. If new classes are reported, we train the new class data along with public dataset, and send the new model weights to global user (Choice 2).

(c) Global update: In this step, if no user reports new classes, we perform label-based averaging using the parameter β, which governs contributions of overlapping labels using corresponding test accuracies (Choice 1). If user reports new classes, we create *Anonymized Data Impressions (DI)* for new classes followed by unsupervised clustering using k-medoids with motivations from [22] (Choice 2).

Typically, statistical heterogeneities are widely observed in practical FL settings, hence Choice 1 handles heterogeneities in local and global update steps [13], while Choice 2 handles new classes in our proposed framework.

3. Experiments and Results

We simulate our experiments using *Raspberry Pi 2* as our user device with **Google Speech Commands (GKWS)** [23] and **UrbanSound8K (US8K)** [2] datasets across 10 FL iterations/communication rounds using our proposed framework. In GKWS, we choose the keywords: Yes, No, Up, Down, Left, Right, On, Off, Stop and Go, and perform regular Mel-frequency Cepstral Coefficients (MFCC) extraction as performed in [1], with sampling frequency of 14400 HZ. The MFCC data is divided into 20 windows and each window is of size 50 ms. US8K, an environmental sound dataset, consists of 10 classes of sound events: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren and street music. We perform similar preprocessing as performed in GKWS for US8K as well.

Public Dataset: We create a Public Dataset (D_0) with 2400 audio frames for GKWS (8 keywords with 300 frames each), and 400 audio frames for US8K (8 sounds with 50 frames each) as shown in Table 1. D_0 is visible to both global and local users in each FL iteration, and is updated with data synthesized for unseen/new classes only – Anonymized Data Impressions.

We initially consider eight labels with an initial Public Dataset in both datasets before streaming new classes (Table 1). We simulate two scenarios for testing just our zero-shot framework – 1) new classes only (homogeneous) with limited users and FL iterations (3 users and 10 FL iterations) for effective analysis of results, 2) new classes with statistical heterogeneities in both labels and models as performed in [13], (10 users and 30 FL iterations). This exhibits near-real-time model heterogeneities as shown in Table 2, and effective convergence.

New Classes: We introduce two new/unseen labels {Stop, Go} for GKWS and {Siren, Street music} for US8K across four FL iterations and two users. In the homogeneous case, for GKWS, we induce 400 samples each with Stop class in iteration 4 for both User 1 and User 2, and 500 samples each with Stop in User 1 iteration 8 and Go in User 2 iteration 8. Similarly, we induce 50 samples each with Siren class in iteration 4 for both User 1 and User 2, and 50 samples each with Siren in User 1 iteration 8 and Street music in User 2 iteration 8. This is the FL scenario with new classes without any heterogeneities. We also discuss similar FL scenarios with statistical heterogeneities.

Table 2: *Heterogeneities in model architectures and new classes changing across FL user iterations for both datasets.*

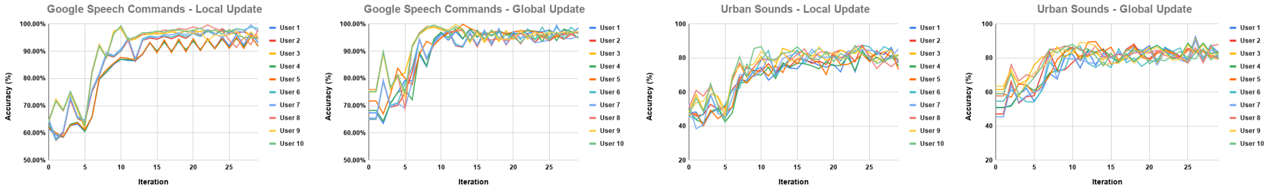
Iteration	New Model	New Class
User 1 Iteration 6	3-Layer ANN (16, 16, 32) ReLU Activation	-
User 1 Iteration 8	1-Layer CNN (16) Softmax Activation	-
User 2 Iteration 4, 6	3-Layer CNN (16, 16, 32) Softmax activation	Stop/Siren
User 3 Iteration 5	4-Layer CNN (8, 16, 16, 32) Softmax activation	-
User 4 Iteration 3, 7	-	Go/Street Music
User 6 Iteration 3, 5	-	Stop/Siren
User 9 Iteration 4	-	Stop/Siren

(a) Label Heterogeneities: In every FL iteration, we consider a random number of audio frames generated between 200-300 samples per label for GKWS, while 40-50 samples per label for US8K. We split these labels across three users such that labels can either be unique or overlapping across users. We also simulate non-IIDness across FL iterations with disparities in both labels and distributions in data (*statistical heterogeneities*).

(b) Model Heterogeneities: We consider the three model architectures as shown in Table 1 motivated from [1, 24], and also change model architectures, filters and activation functions across FL iterations in addition to label heterogeneities with new classes (Table 2). The user iterations are chosen at random.

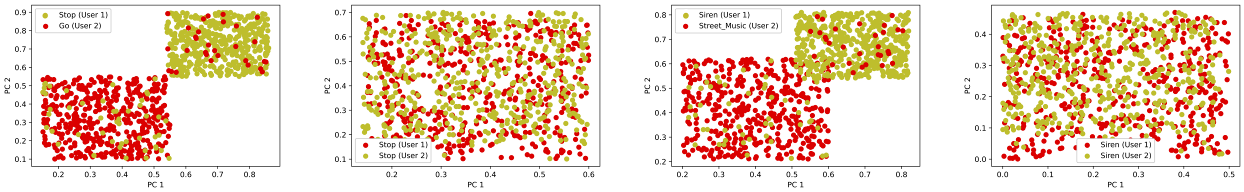
3.1. Discussion on Results

From Table 3, we can observe that there is an accuracy increase in the FL scenario with just new classes (without heterogeneities) in corresponding global updates for all three users than their respective local update accuracies for both datasets in spite of new classes streaming in. The average local-global



(a) (b) (c) (d)

Figure 2: *Local-Global update accuracies (%) across 10 users and 30 FL iterations with new classes and heterogeneities. (a) & (b) show local and global update accuracies respectively for GKWS, while (c) & (d) show these for US8K.*



(a) (b) (c) (d)

Figure 3: *PCA visualizations (with 2 dimensions) after k-medoids clustering with new classes (could be same or different across user devices). (a) & (b) show PCA with different and same classes respectively for GKWS, while (c) & (d) show these for US8K.*

Table 3: *Local-global update accuracies (%) across 3 users and 10 FL iterations, with new classes and without heterogeneities.*

	GKWS			US8K		
User	**Local**	**Global**	**Increase**	**Local**	**Global**	**Increase**
User 1	89.684	93.166	3.482	76.526	80.214	3.688
User 2	91.888	95.28	3.391	75.272	77.944	2.672
User 3	91.517	94.727	3.211	77.61	81.838	4.228
Average	**91.03**	**94.391**	**3.361**	**76.469**	**80**	**3.529**

Table 4: *Final accuracies (%) for both datasets across 10 users and 30 FL iterations, with new classes and heterogeneities.*

Update	**GKWS**	**US8K**
Local	92.5	78.24
Global	96.541	82.498
Increase	**4.041**	**4.258**

accuracy increase across all 10 FL iterations and 3 users is ∼3.361% and ∼3.529% for GKWS and US8K respectively. Similarly, we can also observe that with our proposed framework, the final global accuracies (with convergence after all FL iterations) even with new classes and heterogeneities are 96.541% and 82.498% (Table 4), which are much higher than their respective local update accuracies. The corresponding local-global update accuracies across 30 FL iterations and 10 users are shown in Figure 2. The class similarity matrix with different classes for GKWS is showcased in Figure 4. We can also infer that the clusters effectively formed after performing k-medoids clustering on the new data impressions are equal to the number of new classes, which are visualized using Principal Component Analysis (PCA) in two-dimensions as shown in Figure 3. The new classes can either be different or same across user devices, and these classes are appropriately mapped to the respective end-user devices. The new labels are then finally added back to the overall label set, while the corresponding averaged data impressions are added to the public dataset.

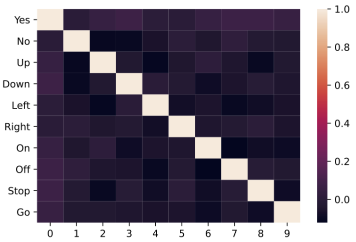


Figure 4: *Class Similarity Matrix for GKWS.*

3.2. On-Device Performance

Raspberry Pi 2 (900MHz quad-core ARM Cortex-A7 CPU with 1GB RAM) is used for evaluating our proposed FL framework as it has similar hardware and software specifications to predominant contemporary IoT/mobile devices. The computation times are identical for both datasets due to similar preprocessing. The training time per epoch for an FL iteration is ∼1.2 sec, while the inference time for one audio sample frame is ∼11 ms. The sizes of the models used are also 520 kB, 350 kB, 270 kB respectively for user architectures mentioned in Table 1.

4. Conclusions

This paper presents a novel framework for handling new labels in a federated learning setting. We propose a zero-shot learning framework by synthesizing Anonymized Data Impressions from Class Similarity matrices to learn new classes across different user devices. We also account for heterogeneities in labels and models across different FL communication rounds, and systematically analyze the results for two widely used audio classification applications – keyword spotting and urban sound classification. We further demonstrate the effectiveness and scalability of our proposed FL framework by simulating our experiments on-device using a Raspberry Pi 2.

5. References

[1] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," *arXiv preprint arXiv:1711.07128*, 2017.

[2] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014, pp. 1041–1044.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "Cnn architectures for large-scale audio classification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 131–135.

[5] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4087–4091.

[6] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Interspeech*, 2015, pp. 1478–1482.

[7] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," in *Proceedings of Machine Learning and Systems*, vol. 1, 2019, pp. 374–388.

[8] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and 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*, vol. 54. PMLR, 2017, pp. 1273–1282.

[9] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, pp. 50–60, 2020.

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

[11] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[12] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[13] G. K. Gudur and S. K. Perepu, "Resource-constrained federated learning with heterogeneous labels and models for human activity recognition," in *Deep Learning for Human Activity Recognition*, vol. 1370, 2021, pp. 57–69.

[14] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6341–6345.

[15] H. Taitelbaum, G. Chechik, and J. Goldberger, "Network adaptation strategies for learning new classes without forgetting the original ones," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3637–3641.

[16] H. Taitelbaum, E. Ben-Reuven, and J. Goldberger, "Adding new classes without access to the original training data with applications to language identification," in *Proc. Interspeech 2018*, 2018, pp. 1808–1812.

[17] A. Hard, K. Partridge, C. Nguyen, N. Subrahmanya, A. Shah, P. Zhu, I. L. Moreno, and R. Mathews, "Training keyword spotting models on non-iid data with federated learning," in *Proc. Interspeech 2020*, 2020, pp. 4343–4347.

[18] G. K. Nayak, K. R. Mopuri, V. Shaj, V. B. Radhakrishnan, and A. Chakraborty, "Zero-shot knowledge distillation in deep networks," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, pp. 4743–4751.

[19] T. Minka, "Estimating a dirichlet distribution," 2000.

[20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[21] G. K. Gudur, B. S. Balaji, and S. K. Perepu, "Resource-constrained federated learning with heterogeneous labels and models," *arXiv preprint arXiv:2011.03206*, 2020.

[22] Z. Shuyang, T. Heittola, and T. Virtanen, "Active learning for sound event classification by clustering unlabeled data," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 751–755.

[23] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[24] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1251–1258.