HETEROGENEOUS ZERO-SHOT FEDERATED LEARNING WITH NEW CLASSES FOR AUDIO CLASSIFICATION

Gautham Krishna Gudur
Global AI Accelerator, Ericsson
gautham.krishna.gudur@ericsson.com

Satheesh Kumar Perepu
Ericsson Research
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 $\sim$4.041% and $\sim$4.258% respectively.

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. (Zhang et al., 2017). Urban environment sound classification is another interesting application particularly in context-aware computing, urban informatics (Salamon et al., 2014). The emergence of deep neural networks have conveniently alleviated problems of creating shallow (hand-picked) features and have achieved state-of-the-art performance in such speech classification tasks (Hinton et al., 2012). With the recent compute capabilities vested in resource-constrained devices, there is a huge research focus on audio classification using on-device deep learning (Chen et al., 2014; Sainath & Parada, 2015).

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 just securely sharing model updates with a server without the need to transfer sensitive local user data (Bonawitz et al., 2019; McMahan et al., 2017). On-device federated learning has been an active area of research addressing challenges on secure communication protocols, optimization, privacy preserving networks, etc. (Li et al., 2020). 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 class 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).

One way of handling model heterogeneities and independence in a federated learning setting is by using knowledge distillation (Hinton et al., 2015) with a common student model architecture on each local device (Li & Wang, 2019). Label and model heterogeneities are handled in an inertial Human Activity Recognition scenario in (Gudur & Perepu, 2021). Federated learning for keyword spotting (Leroy et al., 2019), and new class learning and identification in various speech recognition settings are addressed in (Taitelbaum et al., 2019; 2018). The paper (Hard et al., 2020) proposes a new augmentation technique to reduce false reject rates and 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 new label identification and similarity detection in a zero-shot manner when heterogeneous label and model distributions exist across various FL iterations and users. To the best of our knowledge, none of the papers discuss new label identification in FL settings with statistical heterogeneities for audio classification.

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

2 Our Approach

In this section, we discuss about the problem formulation of new classes and heterogeneities in FL, and our proposed framework (Algorithm 1). The overall architecture is given in Appendix A.

2.1 Problem Formulation:

We assume the following scenario in federated learning. Suppose there are M nodes (devices) in the FL network, holding data with distinct private local data $\mathcal{D}_i = \{x_{i,j}, y_{i,j}\}$ where i is 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 (Li & Wang, 2019) 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 are the unique labels of overall label-set. 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 identify new classes 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 (Nayak et al., 2019) to compute *Anonymized Data Impressions*. 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 (Minka, 2000). *CSM* 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 penultimate layer to the nodes of the classes (Nayak et al., 2019). The Class Similarity Matrix is constructed as,

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

Algorithm 1 Our Proposed Framework

Input: Public Dataset $\mathcal{D}_0\{x_0, y_0\}$, Private Datasets $\mathcal{D}_m^i$, Total users M, Total iterations I, LabelSet l_m for each user, Overall Public LabelSet Y,
Output: Trained Model scores f_G^I
Initialize $f_G^0 = \mathbf{0}$ (Global Model Scores)
for $i = 1$ **to** I **do**
  for $m = 1$ **to** M **do**
    Build: Model $\mathcal{D}_m^i$ and predict $f_{\mathcal{D}_m^i}(x_0)$
    Local Update:
    Choice 1: New classes are not reported
    $f_{\mathcal{D}_m^i}(x_0) = f_G^I(x_0^{l_m}) + \alpha f_{\mathcal{D}_m^i}(x_0)$, where $f_G^I(x_0^{l_m})$ are global scores of l_m with m^{th} user, $\alpha = \frac{len(\mathcal{D}_m^i)}{len(\mathcal{D}_0)}$
    Choice 2: New classes are reported
    Train a new model with $\mathcal{D}_0$ and $\mathcal{D}_m^i$ (new data) together, and send weights of the last layer ($\mathbf{W}_m^i$) to global user.
  end for
  Global Update:
  Choice 1: No user reports new classes
  Update label wise
  $f_G^{i+1} = \sum_{m=1}^{M} \beta_m f_{\mathcal{D}_m^i}(x_0)$, where
  $\beta = \begin{cases} 1 & \text{If labels are unique} \\ \text{acc}(f_{\mathcal{D}_m^{i+1}}(x_0)) & \text{if labels are not unique} \end{cases}$
  where $\text{acc}(f_{\mathcal{D}_m^{i+1}}(x_0))$ 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 $\mathbf{W}_m^i$ (Section 2.2). Average *DI* of all users with new classes, $\mathbf{X}^i = \sum_{m \in M_{S_k}} \mathbf{X}_m^i$, 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

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 $\mathbf{Y}^k = [\mathbf{y}_1^k, \mathbf{y}_2^k, \cdots, \mathbf{y}_N^k] \in \mathcal{R}^{K \times N}$ 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 $\mathbf{Y}^k$,

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

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 (Abadi et al., 2016) for all our experiments.

2.3 Proposed Framework

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

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

(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 (Gudur et al., 2020), where α governs the contributions of new and old models across FL iterations. If new classes are reported, we train the new class data along with public dataset, and send the new model weights to global user.

(c) Global update: In this step, if no user reports new classes, we perform label-based averaging using the parameter β, which governs weightage of overlapping labels using corresponding test accuracies. If user reports new classes, we create *Anonymized Data Impressions (DI)* for new classes followed by unsupervised clustering using k-medoids with motivations from (Shuyang et al., 2017) (Algorithm 1 Choice 2).

Typically, statistical heterogeneities are widely observed in practical FL settings, hence Choice 1 handles heterogeneities in local and global update steps (Gudur & Perepu, 2021), 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)** (Warden, 2018) and **UrbanSound8K (US8K)** (Salamon et al., 2014) datasets (Appendix B) across different FL iterations/commumnication rounds using our proposed framework.

Public Dataset: We create a Public Dataset (D_0) with 2400 audio frames for GKWS (8 keywords with 300 each), and 400 audio frames for US8K (8 sounds with 50 sounds) as shown in Figure 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.

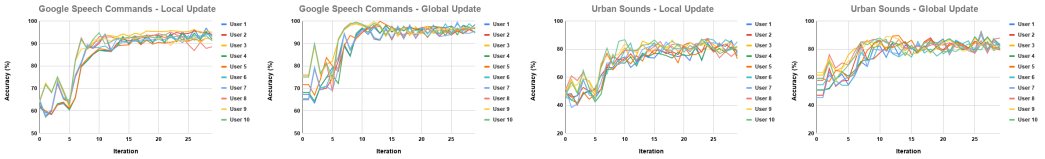
We initially consider eight labels with the initial Public Dataset in both datasets before streaming new classes (Table 1). We simulate two scenarios for testing our zero-shot framework - 1) new classes only (homogeneous) with limited users and FL iterations (3 users and 10 iterations) for effective analysis of results, 2) new classes with statistical heterogeneities in both labels and models as performed in Gudur & Perepu (2021), with more users and FL iterations (10 users and 30 iterations) for effective convergence. This exhibits near-real-time statistical heterogeneities as shown in Appendix C Table 3.

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 class 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.

(a) Label Heterogeneities: In every FL iteration, we also 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 (Zhang et al., 2017; Chollet, 2017), and also change model architectures, filters and activation functions over FL iterations in addition to label heterogeneities with new classes (Appendix C Table 3). The FL user iterations for such heterogeneities were chosen at random.

3.1 Discussion on Results



(a) GKWS - Local Update (b) GKWS - Global Update (c) US8K - Local Update (d) US8K - Global Update

Figure 1: Local-Global update accuracies across 10 users and 30 FL iterations for both datasets with new classes and heterogeneities.

Table 2: Final local-global update accuracies (%) with new classes across users and FL iterations.

(a) 3 users, 10 FL iterations 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**

(b) 10 users, 30 FL iterations with heterogeneities.

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

From Table 2a, we can observe that there is an accuracy increase in FL scenario with just new classes (without heterogeneities) in corresponding global updates for all three users than the respective local update accuracies for both datasets in spite of new classes streaming in. The average local-global accuracy increase across all 10 FL iterations and 3 users is $\sim$3.361% and $\sim$3.529% respectively for GKWS and US8K. 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 2b) which are much higher than their respective local update accuracies. The corresponding local-global update accuracies across 30 iterations are shown in Figure 1. The class similarity matrix of different classes for GKWS is showcased in Appendix D Figure 3, which elucidates the misclassifications. We can also infer that the clusters effectively formed with k-medoids are equal to number of new classes, which are visualized using Principal Component Analysis (PCA) in two-dimensions. The new classes can either be different or same across user devices (Appendix E Figure 4), and these classes are correctly mapped to the respective end-user devices. The new labels are then finally added to the overall label set while the corresponding averaged data impressions are added to the public dataset. Further, Raspberry Pi 2 performance metrics are observed in Appendix F, showcasing the effectiveness of our proposed FL framework.

4 Conclusion

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 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.

REFERENCES

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, 2016.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. In *SysML 2019*, 2019.

Guoguo Chen, Carolina Parada, and Georg Heigold. Small-footprint keyword spotting using deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4087–4091. IEEE, 2014.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Gautham Krishna Gudur and Satheesh Kumar Perepu. Resource-constrained federated learning with heterogeneous labels and models for human activity recognition. In *Deep Learning for Human Activity Recognition*, pp. 57–69. Springer Singapore, 2021.

Gautham Krishna Gudur, Bala Shyamala Balaji, and Satheesh K Perepu. Resource-constrained federated learning with heterogeneous labels and models. *arXiv preprint arXiv:2011.03206*, 2020.

Andrew Hard, Kurt Partridge, Cameron Nguyen, Niranjan Subrahmanya, Aishanee Shah, Pai Zhu, Ignacio Lopez Moreno, and Rajiv Mathews. Training keyword spotting models on non-iid data with federated learning. In *Proc. Interspeech*, pp. 4343–4347, 2020.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, pp. 82–97, 2012.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau. Federated learning for keyword spotting. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6341–6345, 2019.

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37:50–60, 2020.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pp. 1273–1282, 2017.

Thomas Minka. Estimating a dirichlet distribution, 2000.

Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, pp. 4743–4751, 2019.

Tara N Sainath and Carolina Parada. Convolutional neural networks for small-footprint keyword spotting. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 1041–1044, 2014.

Zhao Shuyang, Toni Heittola, and Tuomas Virtanen. Active learning for sound event classification by clustering unlabeled data. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 751–755, 2017.

Hagai Taitelbaum, Ehud Ben-Reuven, and Jacob Goldberger. Adding new classes without access to the original training data with applications to language identification. In *INTERSPEECH*, pp. 1808–1812, 2018.

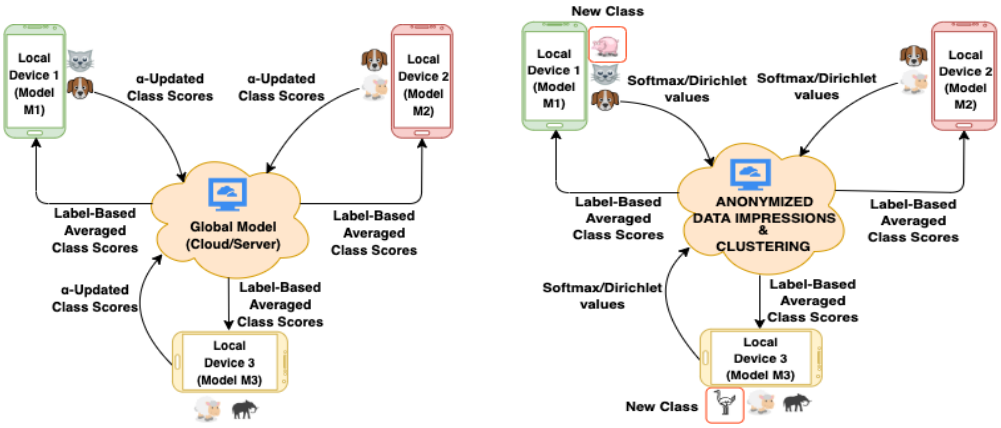
Hagai Taitelbaum, Gal Chechik, and Jacob Goldberger. Network adaptation strategies for learning new classes without forgetting the original ones. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3637–3641, 2019.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra. Hello edge: Keyword spotting on microcontrollers. *arXiv preprint arXiv:1711.07128*, 2017.

A Appendix: Overall Architecture

The overall architecture of our proposed framework of new classes identification in a zero-shot manner in FL settings with heterogeneities, along with existing FL scenarios with only heterogeneities (Gudur et al., 2020) is elucidated in Figure 2.



(a) FL with existing classes

(b) FL with new classes

Figure 2: Overall Architecture of FL with existing classes and proposed framework with new classes. Each local device consists of heterogeneous sets of labels and models, and they interact with the global model (cloud/server). When new labels stream in the devices, our proposed zero-shot FL framework in Figure 2b is triggered, else the conventional FL setting in Figure 2a is triggered. The updated consensus is finally distributed across local models and the process continues.

B Appendix: Datasetd and Dataset Preprocessing

Google Speech Commands (GKWS) Warden (2018) consists of audio clips of one second and one keyword each by thousands of different people. 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 (Zhang et al., 2017), with sampling frequency of 14400 HZ. The MFCC data is divided into 20 windows and each window is of size 50 ms.

UrbanSound8K (US8K) (Salamon et al., 2014), 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. All the sounds in the dataset are urban field-recordings. We perform similar preprocessing using MFCC as previously performed in GKWS for US8K as well.

C Appendix: Model Heterogeneities across Iterations

The model and label heterogeneities across and within different FL iterations are observed in Table 3. Changing model architectures, filters and activation functions over FL iterations exhibit near-real-time model heterogeneities. In addition, we also add label heterogeneities with new classes across different FL user iterations.

Table 3: Details of heterogeneities - model architectures (filters) and new classes changing across FL iterations and users 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 5, 3	-	Stop/Siren
User 9 Iteration 4	-	Stop/Siren

D Appendix: Class Similarity Matrix

The Class Similarity Matrix calculated from Section 2.2 for the 10 classes of Google Speech Commands Dataset (GKWS) is showcased in Figure 3.

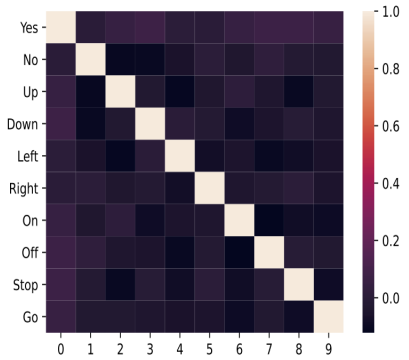
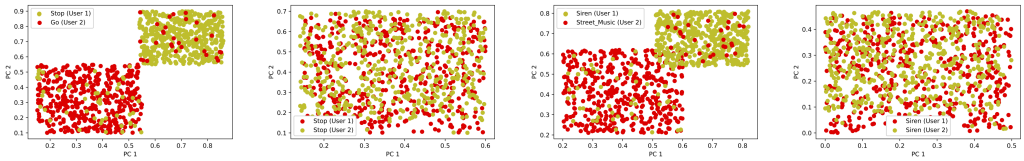


Figure 3: Class Similarity Matrix for Google Speech Commands Dataset.

E APPENDIX: USER REPORTS ON CLUSTERING OF NEW CLASSES

k-medoids unsupervised clustering is performed, and the PCA results (with 2 dimensions) of using new classes which are different across user devices, and also same across user devices are observed in Figure 4. The new classes considered in our experiments are {Stop and Go} for Google Speech Commands dataset, and {Siren and Street music} for UrbanSound8K dataset. The number of clusters returned are the new classes which are correctly mapped to respective end-user devices, and the new labels are added to the overall label set and corresponding data impressions are added to the public dataset. This process is repeated for creating further anonymized data impressions.



(a) GKWS - Different Class (b) GKWS - Same Class (c) US8K - Different Class (d) US8K - Same Class

Figure 4: PCA (with 2 dimensions) of k-medoids unsupervised clustering with new classes, with same and different classes for both datasets.

F APPENDIX: 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 (HW/SW) specifications to predominant contemporary IoT/mobile devices. The computation times are identical for both datasets due to similar preprocessing. The size of the models used are also 520 kB, 350 kB, 270 kB respectively for user architectures mentioned in Table 1.

Table 4: Computation Times with Raspberry Pi 2

Process	**Time**
Training time per epoch in an FL iteration (i)	$\sim$1.2 sec
Inference time	$\sim$11 ms