2022 10th International Conference on Affective Computing and Intelligent Interaction (ACII) | 978-1-6654-5908-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ACII55700.2022.9953856

Federated Learning for Affective Computing Tasks

Krishna Somandepalli, Hang Qi, Brian Eoff, Alan Cowen, Kartik Audhkhasi, Josh Belanich and Brendan Jou

Abstract—**Federated learning mitigates the need to store user data in a central datastore for machine learning tasks, and is particularly beneficial when working with sensitive user data or tasks. Although successfully used for applications such as improving keyboard query suggestions, it is not studied systematically for modeling affective computing tasks which are often laden with subjective labels and high variability across individuals/raters or even by the same participant. In this paper, we study the federated averaging algorithm `FedAvg` to model self-reported emotional experience and perception labels on a variety of speech, video and text datasets. We identify two learning paradigms that commonly arise in affective computing tasks: modeling of self-reports (*user-as-client*), and modeling perceptual judgments such as labeling sentiment of online comments (*rater-as-client*). In the user-as-client setting, we show that `FedAvg` generally performs on-par with a non-federated model in classifying self-reports. In the rater-as-client setting, `FedAvg` consistently performed poorer than its non-federated counterpart. We found that the performance of `FedAvg` degraded for classes where the inter-rater agreement was moderate to low. To address this finding, we propose an algorithm `FedRater` that learns client-specific label distributions in federated settings. Our experimental results show that `FedRater` not only improves the overall classification performance compared to `FedAvg` but also provides insights for estimating proxies of inter-rater agreement in distributed settings.**

Index Terms—**emotion experience, emotion perception, federated learning, sentiment classification**

I. Introduction

In recent years, there is growing interest in employing machine learning to model affective computing (AC) constructs such as emotion and sentiment. AC tasks are often subjective, personal and private in nature. Due to the sensitive nature of the data associated with these attributes and tasks, minimizing the need to centrally store such data is a much desired feature. The recent advances of federated learning offer one solution here to advance user agency over their data, labels, and compute.

Federated learning (FL) [1] has greatly enabled large-scale distributed learning without the sharing of user data with a central datastore – enabling opportunities for preserving privacy, and even security, when potentially sensitive data from users is used to train machine learning models. In FL, a group of user client devices (often referred simply as *clients*) collaborate with a server to train statistical models with data stored only locally (on-device). Besides many proof-of-concept experiments such as handwritten digit or image classification, keyword spotting and visual question answering [2], [3], FL was successfully applied in large-scale tasks such as next-word prediction on mobile keyboards [4], improving on-device speaker verification [5] and speech recognition [6]. While these underscore the promise of FL for developing distributed machine learning models at scale, FL is not systematically studied for AC tasks. Existing studies at the intersection of FL and AC (e.g., [7], [8]) treat labels for AC tasks as objective groundtruth for training models. They do not study the variability associated with the subjective labels obtained from raters. Our paper addresses this gap through carefully considering how the data/labels in an AC task maybe collected in a federated setting.

Corresponding author: Krishna Somandepalli, ksoman@google.com
This work was done by Alan Cowen while a visiting researcher at Google.

To understand the applicability of FL beyond the data provenance and computational constraint considerations, two key design questions are crucial: *data partition: who are the clients (population of federated users)?* and *what is the nature of data and labels needed for the learning task?* In exploring these questions for AC tasks, we identify two distinct paradigms: (1) *user-as-client* and (2) *rater-as-client*. Consider training a model to predict a user's experienced emotion from selfies on their device. Here, we consider the users as individual clients, and call this learning paradigm *user-as-client*. While some prevalent FL applications such as next-word prediction perform data partition per *user-as-client*, the associated learning task is not highly subjective as in AC tasks. Examples that can be conceptualized in this paradigm include self-reported emotion from face images [9] or speech [10]. The data/labels are stored only on the client's device, promoting user trust critical for such sensitive attributes. Consider another example of asking multiple users to label their response to the same stimulus. With FL, modeling these raters as clients can ensure the server side does not have direct access to the data being labeled as well as the labels. We call these settings *rater-as-client*. Examples include classifying sentiment perceived from comments on Reddit [11] and reporting likes/dislikes to sensitive media content.

The objective in this work is to empirically assess the performance of FL in AC tasks for the two paradigms. Specifically, we study classifying self-reported emotional experiences in a *user-as-client* paradigm and emotional perceptual judgments in a *rater-as-client* setting on speech, video and text datasets. We first discuss the two proposed FL paradigms and highlight the benefits, expectations and challenges in contrast to non-federated (centralized) model training where data and labels are available on a central datastore. We next describe our proposed federated modeling of raters as clients followed by a series of experiments to evaluate classification performance of FL with respect to centralized models.

Authorized licensed use limited to: GOOGLE. Downloaded on April 01,2023 at 14:53:22 UTC from IEEE Xplore. Restrictions apply.

II. Related Work

Perhaps the most widely used FL backbone is the federated averaging (`FedAvg`) algorithm [1]. Here, a variable number of devices participate in training local (on-device) models for a small number of epochs and communicate the weights to a central server that aggregates them and returns the updated weights to available devices. In real-world scenarios, many problems in FL stem from non-IID distributions of data resulting from *systems heterogeneity* (variability of device characteristics and availability of devices, a.k.a. stragglers) and *statistical heterogeneity* (two devices may not have the same distributions of data/labels for a learning task), beyond communication and on-device computation costs [3]. To address some of these challenges, [12] proposed two algorithms: `FedIR` (IR: importance resampling) to address class imbalance and `FedVC` (VC: virtual clients) to tackle variability between devices. More recently, the problem of stragglers was addressed in `FedProx` [2] where the on-device gradient updates are modified by measuring their dissimilarity with server updates.

The use of FL for AC tasks is still nascent. Notably, the *Sentiment140* dataset [13] was used to benchmark sentiment classification in tweets using `FedAvg` [14] and `FedProx` algorithms. The positive/negative sentiment labels here were automatically determined by the emoticons used by the Twitter users. Speech emotion recognition was studied using the IEMOCAP dataset in [7] and [8]. They used the emotion labels pre-aggregated using majority vote to train models using FL. However, a systematic study of the self-reported emotion labels or those collected from multiple raters is lacking in related work. To this end, we discuss the two FL paradigms, that arise when considering the nature of labeling and users involved in the collection of an AC dataset.

A. User-as-client paradigm

Let us revisit the example of modeling a user's experienced emotion reports from selfies on their mobile device. In a federated setting, the user data and labels from client devices remain local to the device, enabling privacy and anonymity. The associated data samples and self-reported labels may be characterized by large intra-individual variability resulting from personal preferences that change over time and could vary across individuals. Although the ultimate objective would be to develop personalized models tailored to each client, a first step is to evaluate how well (classification performance) a federated model performs versus a non-federated (centralized) one for a fixed client pool. The potentially large intra- and inter-individual variability of the data and self-reports give rise to a high statistical or data heterogeneity [1] in training federated models. While data heterogeneity is common in other applications (e.g., object classification), AC tasks elicit labels that are particularly subjective and the "ground truth" is either absent or unknown [16].

We highlight two recent works in this domain to delineate the concept of subjective labels from noisy labels. [17] examined the performance of `FedAvg` with simulated noisy labels in CIFAR-10 [18] and a large-scale dataset of clothing with noisy labels obtained from online shopping websites [19] whereas [20] explored federated representation learning for human activity recognition. Although these studies addressed heterogeneity resulting from noisy labels, the learning tasks have a known "ground truth". Furthermore, the FL simulations were performed by randomly dividing the data into client subsets which ensured that the label noise is uniformly distributed across all clients. In contrast, when the data are partitioned into *user-as-client* for learning from self-reports, the associated subjectivity introduces statistical noise *specific* to each client, inflating data heterogeneity.

B. Rater-as-client paradigm

When self-reports are not available, a widely adopted practice for modeling human-perception tasks is to obtain labels from multiple people other than self (*raters*). Due to the context- and person-dependent nature of these tasks, true "ground truth" is often absent or unknown. Typically, in non-federated model training, some form of *label aggregation* is performed to obtain a single set of labels for model supervision (e.g., [21]). To apply FL in this context, we propose having each rater act as a client and partition the data accordingly. Modeling *rater-as-client* ensures that the central server is not only agnostic to the identity of the samples rated by the client but also to the identity of the client associated with their labels. While this promotes user trust and anonymity (for example, a 2012 study [22] showed that likes/dislikes from users can be used to infer their personal traits), it presents two distinct challenges for the application of FL.

First, two raters may respond to the same input stimulus differently leading to *concept heterogeneity*, i.e., the distribution of local data on two devices could be identical but the distribution of the labels conditioned on the data are not identical. With the notable exception of [23], concept heterogeneity is not widely studied in federated learning. [23] used FL for classifying symptom severity in patients with Parkinson's disease by measuring hand tremors while holding a smartphone using labels collected from two expert raters across three hospitals. The data were partitioned by treating different hospitals as individual clients and the federated models were trained with labels from the first expert and evaluated using labels from the second expert. Unlike this experiment — which limits the concept heterogeneity introduced by partitioning hospitals (and not raters) as clients — it may not be feasible to collect labels for AC tasks from a fixed pool of raters uniformly. This is largely due to the practical limitation of intermittent availability of devices participating in federated training. Unlike symptom severity, labeling in AC tasks involves human perceptual judgment and lacks objective labels. In such cases, aggregated labels (e.g., majority vote) are often treated as a proxy for the ground truth. Thus, the goal of a federated model in this context is to not learn how an individual rater may respond to a stimulus, but rather to approximate the "average" perceptual response of a rater population.

Authorized licensed use limited to: GOOGLE. Downloaded on April 01,2023 at 14:53:22 UTC from IEEE Xplore. Restrictions apply.

Dataset	Modality	Classification task	No. samples	No. ratings	Clients	No. clients	No. samples/client (min — max)	No. classes
IEMOCAP [10]	Speech	Self-report	2,409	2,409	users	8	53 — 633	10
		Perception rating	4,784	14,352	raters	6	170 — 4,709	10
BRAVE [15]	Video	Self-report	49,102	49,102	users	1,399	1 — 813	34
		Perception rating	308,572	649,900	raters	3,286	25 — 2,575	42
GoEmotions [11]	Text	Perception rating	58,009	207,813	raters	81	7 — 10,508	28

TABLE I
DESCRIPTIVE STATISTICS OF DATASETS STUDIED IN *user-as-client* AND *rater-as-client* FEDERATED SETTINGS.

Second, direct label aggregation is impossible in FL as the identity of data/labels rated by the clients is not visible to the server model. To address this challenge, we look to hidden rater models proposed in the non-federated domain [24], [25]. Here, the conditional distribution of a label for each rater is estimated with respect to the "true" underlying distribution of the corresponding label which is the unknown ground-truth in the case of subjective labels. A variant of this general idea was recently extended to discrete multi-class classification problems in [26] by estimating rater-specific confusion matrices. Inspired by these ideas, we propose an algorithm called `FedRater` to model raters in a federated setting by jointly estimating client-specific stateful parameters while minimizing the overall loss of the shared model.

III. DATASETS

Here, we describe datasets studied in this paper associated with the two AC tasks of classifying self-report and perception ratings. These datasets are of different modalities (audio, text and video) labeled in a multi-label fashion, along with the availability of (anonymized) participant and rater ID; allowing us to simulate *user-as-client* and *rater-as-client* settings. The descriptive statistics of the datasets and related tasks are presented in Table I.

A. Interactive Emotional Dyadic Motion Capture

IEMOCAP, developed at the University of Southern California (USC) [10], is widely used for benchmarking emotion-related classification. It consists of 10 professional actors (five women and five men) either enacting a script or improvising (improv). The data are organized into five sessions where each session includes interactions between a female and male actor. Each sample is an *utterance* corresponding to manually segmented speaker turns from interactions with labels along 10 discrete emotion classes. In our paper, we only focus on the improv part of IEMOCAP as it has labels from both the participants themselves (self-report, for user-as-client setting) as well as labels from multiple raters (perception ratings, for rater-as-client setting). Only four out of five improv sessions have self-report, resulting in about 2400 samples across eight clients for simulating *user-as-client* whereas all five sessions were labeled by three raters per utterance from a pool of six USC students. This resulted in a total of about 14K samples across six clients for simulating *rater-as-client*. The number of utterances and corresponding labels are described in Table I.

Features: Utterance-level acoustic features, i.e., functionals applied to low-level acoustic descriptors were extracted from openSMILE [27]. Specifically we extracted 88 features using the "eGemaps" configuration provided in openSMILE. They were based on the Geneva Minimalistic Acoustic Parameter Set (Gemaps, [28]), which was developed for benchmarking AC tasks in voice research to minimize differences caused by varying parameter sets or their implementations. These features are widely used for emotion-related classification tasks [29].

B. Berkeley Reactions to Affective Video Elicitors

BRAVE [15] is a recent corpus of over 49K video recordings from nearly 1400 participants reacting to a pre-selected set of short evocative video stimuli. The participants label their emotional response to the stimulus along 34 discrete emotion classes and record their facial expressions while reacting to the video. We used the reaction videos (mean duration=12.2s) from the participants as input and self-report labels for simulating *user-as-client*. The reaction videos were then segmented into non-overlapping clips of 2s duration and labeled by multiple external raters along 42 discrete emotion classes, providing perception ratings. Each clip has labels from at least two raters resulting in nearly 650K samples across 3286 clients for simulating *rater-as-client* (See Table I for details).

Features: Consistent with prior work in facial expression research [9], [30], we extract face-based features using the NN2-FaceNet architecture [31] at 6 FPS. Specifically, we apply average pooling to the inception (5a) block with a 7×7 feature map composed of 1024 channels resulting in a multi-dimensional feature vector per detected face.

C. GoEmotions

GoEmotions [11] is a corpus of about 58K carefully curated comments extracted from Reddit with 28 discrete emotion labels from 82 human raters. GoEmotions does not have self-reported emotion labels. Thus, we only simulate *rater-as-client*. For centralized model training, we use the subset of the data with majority agreement labels released by [11] Each sample was labeled by 2–5 raters with an average of 3 raters. We only retain raters who labeled more than one sample resulting in a total of about 207K samples from 81 clients to simulate *rater-as-client* training.

Features: As described in the GoEmotions paper [11], we extracted a 768-dimensional dense feature embedding from the BERT-base model [32].

Authorized licensed use limited to: GOOGLE. Downloaded on April 01,2023 at 14:53:22 UTC from IEEE Xplore. Restrictions apply.

IV. Methods

In this section, we first describe FedAvg [1] which we use to train a baseline FL model in all experiments. Next, we discuss a probabilistic model of multiple raters for centralized training and present our proposed adaptation for FL.

A. *FedAvg: Federated averaging*

The pseudocode for FedAvg is detailed in Algorithm 1 in black font. For every federated round t on the server model (**Server training loop**), K clients referred to as *report goal* are randomly selected from a pool of available clients, (Line 4). All selected clients $k = 1, \dots, K$ receive the same starting model θ_t from the central server and perform local mini-batch SGD optimization with a learning rate η for E epochs (Line 16–17) over n_k samples $\mathbf{X_k}$ available on-device. The accumulated model updates for each client $\Delta\theta^k$ (Line 22) are communicated back to the server, where a weighted aggregation of the updates across all clients is performed (Line 10) with weights n_k/n proportional to the client sample size n_k. The federated training round is completed by updating the starting model with the aggregated updates with a server learning rate ν (Line 11). In all our experiments, the SGD optimizer on the local models is replaced by a momentum optimizer, which was shown to improve robustness to non-identically distributed client data [33].

Algorithm 1: FedAvg, FedRater, FedRater+

```
1  Server training loop;
2  Initialize $\theta_0$
3  for each round $t = 0, 1, \dots$ do
4      Select $K$ clients sampled uniformly
5      for each client $k = 1, 2, \dots, K$ do
6          $n_k = |\mathbf{X}_k|$
7          $\Delta\theta_t^k \leftarrow$ ClientUpdate($k, \theta_t$)
8      end
9      $n = \sum_{k=1}^{K} n_k$
10     $\bar{\mathbf{g}}_t \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \Delta\theta_t^k$
11     $\theta_{t+1} \leftarrow \theta_t - \nu\bar{\mathbf{g}}_t$
12 end
13 ClientInitializer($k, c$):                    // $c$ classes
14 Initialize $\hat{\mathbf{A}}_k = \mathbf{I} \in \mathbb{R}^{+c \times c}$, $\hat{\alpha}_k = 1 \in \mathbb{R}^{+}$
15 ClientUpdate($k$, $\theta_t$):
16 $\theta \leftarrow \theta_t$
17 for each local mini-batch $b$ over $E$ epochs do
18     $\mathcal{L}(b;\theta) = \sum_{i=1}^{b} \mathbf{CE}(\hat{\mathbf{A}}_k \hat{p}_\theta(\mathbf{x_i}), y_i^{(k)}) + \lambda \text{Tr}(\hat{\mathbf{A}}_k)$
19     $\mathcal{L}(b;\theta) = \sum_{i=1}^{b} \hat{\alpha}_k \mathbf{CE}(\hat{\mathbf{A}}_k \hat{p}_\theta(\mathbf{x_i}), y_i^{(k)}) + \lambda \text{Tr}(\hat{\mathbf{A}}_k)$
20     $\theta \leftarrow \theta - \eta\mathcal{L}(b;\theta)$
21 end
22 return $\Delta\theta \leftarrow \theta_t - \theta$ to server
```

B. *FedRater: Modeling raters in a federated setting*

Let us consider K raters and denote the label distribution of each rater $k = 1, \dots, K$ for a given sample $\mathbf{x}$ as $p(y^{(k)}|\mathbf{x})$, then by assuming that the raters are statistically independent, the joint label distribution across all raters can be written as

$$p(y^{(1)}, \dots, y^{(K)}|\mathbf{x}) = \prod_{k=1}^{K} \int_y p(y^{(k)}|y, \mathbf{x}) \cdot p(y|\mathbf{x}) dy \quad (1)$$

where $p(y|\mathbf{x})$ is the "true" label distribution and $p(y^{(k)}|y, \mathbf{x})$ describes the rater-specific model for sample $\mathbf{x}$. By assuming that the label noise introduced by raters is independent of $\mathbf{x}$, the rater-specific distribution can be parameterized as $p(y^{(k)} = i|y = j, \mathbf{x}) = \alpha_{ij}^{(k)}$ where i, j correspond to the label values of the rater and true distribution respectively. [26] extended this probabilistic model to a multi-class classification problem by estimating a rater-specific confusion matrix $\hat{\mathbf{A}} \in \mathbb{R}^{+c \times c}$ with c number of classes. The modified classification loss with cross-entropy loss **CE** is:

$$\mathcal{L}(b;\theta) = \sum_{i=1}^{b} \mathbf{CE}(\hat{\mathbf{A}}_k \hat{p}_\theta(\mathbf{x_i}), y_i^{(k)}) + \lambda \text{Tr}(\hat{\mathbf{A}}_k) \quad (2)$$

where $\hat{p}_\theta(\mathbf{x}_i)$ and $y_i^{(k)}$ denote the model prediction and the labels corresponding to sample $\mathbf{x}_i$, and $\text{Tr}(\cdot)$ is the trace operation applied on the rater-specific confusion matrix with a corresponding weighting factor λ. The trace factor in the loss ensures that the $\hat{\mathbf{A}}$ matrices are diagonally dominant.

Our proposed adaptation of the rater modeling for a federated setting is described in blue font and referred to as FedRater in Algorithm 1. Similar to FedAvg, K clients are randomly selected from the pool of available raters (Line 4). When a client first receives a starting model θ_t, a rater-specific confusion matrix $\hat{\mathbf{A}}$ is initialized (Line 14, in blue) and added to set of local model parameters on the client side which are updated as the model is trained for E local epochs (Line 18). This parameter is stateful. It is updated when θ_t is received from the server model and a client is available and selected by the server. Furthermore, $\hat{\mathbf{A}}$ is only available locally on the client's device. It is not visible to the server. FedRater does not change the server training loop and can be easily incorporated into FedAvg.

The estimated confusion matrix, along with trace normalization ensures that the distribution of rater-specific predictions are close to the "true" distribution of the labels. In a centralized model training, the "true" distribution in often estimated using some form of label aggregation (weighted sum). For example, when the labels are binary, a median operation across the available ratings per sample gives the majority vote label distribution. However, for training federated models, we do not have access to all the ratings per sample. In this context, we pose the label aggregation operation (weighted sum across all ratings) as a task of learning rater-specific scaling of the loss function using the parameter $\alpha \in \mathbb{R}^{+}$. We refer to this algorithm as FedRater+ (shown in red, Algorithm 1). Similar to FedRater when a client k first receives a set of updates from the server model, two stateful parameters α_k and $\mathbf{A}_k$ are initialized (Line 14), and updated when training locally on-device using a modified loss function as shown in Line 19 of Algorithm 1. It is important to note that α_k is

Authorized licensed use limited to: GOOGLE. Downloaded on April 01,2023 at 14:53:22 UTC from IEEE Xplore. Restrictions apply.

separate from the weighting factor n_k/n used for gradient updates (Line 10). Intuitively, α_k is analogous to the weighting factor used for label aggregation in centralized model training, measuring a proxy of "rater quality". In case of subjective labels, rater quality can be approximated by measures of inter-rater agreement, described in the next section.

C. Class-normalized inter-rater agreement (c-IRR)

When a rater is allowed to choose more than one label, raw IRR per sample such as Cohen's Kappa and Krippendorf's alpha do not control for the base rates of choosing a certain class. We must normalize for the number of times that a rater chooses a class label among all the samples rated by the given rater. We refer to this measure as class-normalized IRR (c-IRR). Let $\mathbf{Y}_k \in \mathbb{R}^{N \times c}$ be a c-class label matrix across N samples labeled by rater $k \in \{1, \ldots, K\}$. For the same sample subset, let $\mathbf{Y}_{k'} \in \mathbb{R}^{N \times c}$ be the label matrix obtained from raters other than k. The c-IRR measure for rater k is computed as:

$$\rho_k = \sigma(vec(\mathbf{Y}_k - \bar{\mathbf{Y}}_k), vec(\mathbf{Y}_{k'} - \bar{\mathbf{Y}}_{k'})) \tag{3}$$

where σ denotes a Pearson correlation coefficient (measure of similarity), and $\bar{\mathbf{Y}}_{\cdot}$ denotes the column-wise mean which estimates the base rate of choosing a class label. When a sample is rated by more than two raters, the labels of the raters in label matrix $\mathbf{Y}_k$ are repeated to match the dimensions of $\mathbf{Y}_{k'}$ matrix, allowing us to compute correlation.

Note that c-IRR can only be obtained for raters who labeled more than one sample, to robustly estimate the base rates of a rater choosing a class label. In real-world settings, c-IRR can only be estimated in centralized and not federated settings. We use c-IRR as a proxy of rater quality to understanding its relationship to the client-specific parameters estimated by `FedRater+` in *rater-as-client* simulations.

V. EXPERIMENTS

The primary objective of this work is to assess how well FL performs for subjective tasks compared to centralized models under the same dataset (train/test splits) and model configuration conditions. In developing centralized models, the goal was not necessarily to have state-of-the-art performance for each dataset but to develop a competitive baseline using features informed by related work. When using the same features and same amount of training data, the performance of machine learning models trained with FL algorithms are inherently upper bound by that of the models trained in the centralized fashion.

A. Model implementation and evaluation setup

Consistent with previous work, for both IEMOCAP (88-dim low-level acoustic features) and GoEmotions (768-dim BERT features), we train models by adding a dense output layer on top of the input features for fine-tuning, with sigmoid cross-entropy loss for multi-label classification.

For videos in the BRAVE dataset, we used a model configuration informed by a previous related work [9]. For self-report classification, where the average duration of videos is about 12s, we used two long short-term memory (LSTM) layers each with 64 recurrent cells. The output of the LSTM is then fed through a mixture of experts (MoE) layer (2 mixtures and a dummy expert) followed by an output layer with sigmoid cross entropy loss. For classifying perception ratings from clips (average duration of 2s), we used two LSTM layers each with 32 recurrent cells followed by a MoE layer with 3 experts and an output layer with sigmoid cross entropy loss. In both cases, the number of LSTM layers, recurrent cells, and experts was tuned for the centralized models, and the same configuration was used for training federated models.

Following hyper-parameter tuning, we used a batch size of 64 for all experiments. In our initial experiments, we found that `FedAvg` algorithm performed better with a momentum optimizer on the server model than a SGD optimizer. Hence for all FL experiments, we used a momentum optimizer (learning rate=0.01, momentum=0.9) on the server model and a SGD optimizer on the client model. A momentum optimizer with the same parameters was used for training centralized models as well.

B. User-as-client setting

As described in the Datasets section, we use the self-reports from IEMOCAP and BRAVE where the participants (users) reporting their emotional experience are treated as clients. All evaluations here are done in an open-set fashion, i.e., the users in test-set are not seen during training. For IEMOCAP, we created four different splits with utterances from six users across three sessions for training/validation, and utterances from two users of the remaining session for testing. For BRAVE, which has a larger number of users, we created five different splits where data of 200 participants was used for testing and that of the remaining participants was used for training. The performance metrics reported are averaged across all train/test splits. In FL simulations, open-set evaluation helps evaluate how well models generalize for unseen participants given that the labels are subjective in nature. Federated models are trained with varying two parameters: number of clients selected in each server training loop, i.e., report goal in Algorithm1 with $K = \{2, 5, 6\}$ for IEMOCAP, and $K = \{2, 5, 10, 20\}$ for BRAVE. The report goal for BRAVE (which has 1199 users in train set) was chosen to simulate a case where only a handful of users are available in each training loop. The number of local epochs E on the client side (See Line 17, Algorithm 1) was tuned over $\{1, 3, 5\}$. We did not observe a significant change in overall model performance by varying E, consistent with observations in related work [12].

C. Rater-as-client setting

We use the rater ID and the corresponding perception ratings from in IEMOCAP, GoEmotions and BRAVE to simulate *rater-as-client* setting. For IEMOCAP, Session04 is used for testing whereas for BRAVE, we use one of the splits from the previous setting. For GoEmotions, we use the splits provided by the authors [11].

Authorized licensed use limited to: GOOGLE. Downloaded on April 01,2023 at 14:53:22 UTC from IEEE Xplore. Restrictions apply.

Per-class AP (%)	IEMOCAP (c=10)	BRAVE (c=34)
`FedAvg` (K=4)	12.9 ± 14.1	11.7 ± 9.4
`FedAvg` (K=6)	**14.8 ± 13.1**	11.9 ± 10.2
`FedAvg` (K≥10)	—	**12.8 ± 10.1**
Centralized	18.4 ± 14.2	13.1 ± 9.2

TABLE II
PERFORMANCE EVALUATION IN *user-as-client* SETTING FOR DIFFERENT REPORT GOAL CHOICES AVERAGED ACROSS ALL TRIALS.

We create two centralized baselines for each dataset
(1) **cez**: Training with aggregated labels as "ground truth"
(2) **cez-cm**: Learning rater-specific confusion matrices while minimizing loss function per Eq. 2 [26].
Label aggregation on all datasets is performed as follows: if there was consensus among the raters on *at least one class*, a majority vote across all labels for each sample is used as the aggregated label, otherwise, the labels across the raters are simply averaged. Due to the subjective nature of the tasks, aggregated labels are considered as "ground truth" for testing.

For federated models, we wish to test if the server model behaves as an "average rater", i.e., the labels predicted by the FL model must be closer to the majority vote labels than any individual rater (client). Thus, we use the aggregated labels in each dataset as the "ground truth" for evaluation. It is important to underscore that unlike centralized model training, two raters may have same data features but have different labels in a federated setting, and the server model does not know the identity of the samples being rated. While the train/test splits are created in an open-set fashion with respect to participants, the raters in the test set are a subset of raters seen during training. This simulates a FL setting where the pool of raters or the client population is fixed and the server model is expected to learn the overall rater behavior, while generalizing to data from unseen participants. For **cez-cm**, `FedRater` and `FedRater+`, the trace normalization weight λ (see Eq. 2) is set 0.001 after parameter tuning. The report goal K was chosen to simulate systems heterogeneity i.e., not all few raters available for each training loop. It was tuned over $K = \{5, 10, 15\}$, except for IEMOCAP with $K = \{2, 4, 5\}$. We chose this paramter set to simulate the case where maximum number of clients available in each round in strictly less than the total number of raters available. This is why the choices for K in this setting are not the same as in *user-as-client* (see 'users' vs. 'raters' counts in Table I, column 7). All federated models are trained for 20K rounds. To compare the performance of multi-label classification tasks, we report per-class average precision (AP) (mean, std), we also monitored micro-averaged AP during evaluation.

VI. RESULTS

We first present the classification results for the two settings with users and raters as clients and discuss their performance vis-à-vis centralized models. We next present `FedRater` model analysis on the BRAVE dataset.

Dataset (No. classes)	IEMOCAP (c=10)	BRAVE (c=42)	GoEmotions (c=28)
c-IRR	0.53 ± 0.08	0.2 ± 0.15	0.55 ± 0.13
Report goal	5	5	10
	Per-class average precision (%)		
`FedAvg`	27.2 ± 21.9	8.7 ± 8.9	18.6 ± 16.8
`FedRater`	27.6 ± 22.2	9.0 ± 7.7	20.8 ± 16.7
`FedRater+`	**29.1 ± 23.2**	**10.2 ± 8.3**	**21.0 ± 17.4**
cez	30.0 ± 23.6	10.9 ± 9.1	20.9 ± 15.2
cez-cm	30.5 ± 23.9	10.5 ± 8.6	22.3 ± 17.1

TABLE III
EVALUATION IN *rater-as-client* SETTING FOR THE BEST PERFORMING REPORT GOAL IN EACH DATASET.

A. User-as-client setting

The per-class AP of centralized models for classifying self-reports in IEMOCAP and BRAVE are presented in Table II. First, in the case of IEMOCAP, the performance of `FedAvg` model improves by increasing the report goal K (about 2% AP), and on average performs about 80% as well as the centralized upper-bound. For IEMOCAP, both centralized and federated models perform the best for emotions with overt low-level markers ("excited", "neutral" and "frustration": AP $> 24\%$). The centralized model performance is consistent with previous reports [29].

For BRAVE, $K > 10$ did not further improve the model performance. Compared to the upper bound, `FedAvg` at $K = 10$ performs at 98%, suggesting that federated models can perform on par with centralized models for learning subjective affective constructs when a large user pool is available (over 1000 users in BRAVE vs. 8 users in IEMOCAP), although the number of users needed for each training loop is minimal ($K \geq 10$). Both centralized and federated models on the BRAVE dataset performed the best for "amusement", "disgust" and "joy" (AP$> 36\%$).

B. Rater-as-client setting

First, we examined the range of rater c-IRR (see Eq. 3) scores across all datasets. As shown in Table III, both IEMOCAP (6 raters) and GoEmotions (81 raters) have moderate to high agreement (c-IRR$\geq$ 0.5), whereas the agreement scores across the 3286 raters in BRAVE are generally low (25–75 percentile range: 0.06 and 0.32, respectively). These c-IRR scores are also reflective of the per-sample majority agreement: 83.1% for IEMOCAP and 100% for GoEmotions vs. 18% for BRAVE. Thus, these datasets have a wide variety of rater agreements, which is common in real-world subjective tasks.

Centralized rater modeling (**cez-cm**) slightly improved the overall performance (except for BRAVE, likely due to low inter-rater agreement) which we use as the upper bound for comparing federated models. In general, we observed that `FedAvg` performs better (i.e., closer to centralized models) when the rater agreement is higher: Relative performance of 90% for IEMOCAP and 88% for GoEmotions vs. 79% for

Authorized licensed use limited to: GOOGLE. Downloaded on April 01,2023 at 14:53:22 UTC from IEEE Xplore. Restrictions apply.

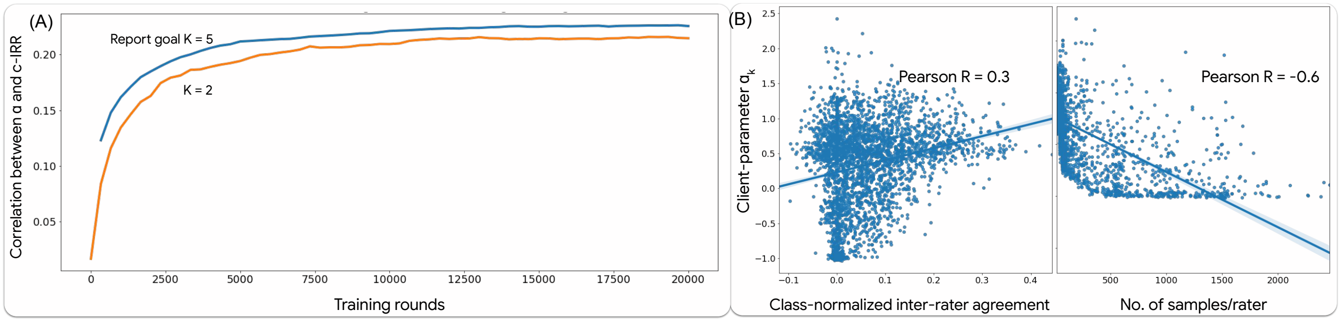

Fig. 1. Analysis of the client-specific α learnt by FedRater+ for BRAVE: (A) Correlation between α and c-IRR during training, and (B) α captures a proxy of c-IRR distinct from the number of samples labeled by the rater .

BRAVE. Across all datasets, we observed a consistent significant improvement (FedRater+ $>$ FedRater $>$ FedAvg: paired t-test on per-class AP scores, p-value$<$ 0.01) in FL performance for FedRater and FedRater+ compared to FedAvg. While the confusion matrix $\hat{\mathbf{A}}_k$ in FedRater and FedRater+ estimates the distribution of rater-specific labels with respect to that of the "true" labels, the additional parameter α_k in FedRater+ adjusts the client updates as they are communicated back to the server (Line 20, Algorithm 1). FedRater+ achieved a further improvement of 1% AP over FedRater. We also observed that a small number of clients per server loop (See report goal row in Table III) is sufficient (increasing K further did not yield significant improvements).

To understand the source of performance gains, we examine AP with respect to agreement scores per class. While FedRater and FedRater+ models not only boost the performance for classes with high agreement, we also noticed a significant improvement for classes with moderate agreement, on par with the upper bound. Detailed analysis are presented in the supplementary material. In order to understand why FedRater+ improves the overall performance, we conducted post-hoc analysis comparing the client-specific parameters in FedRater+ to per-rater c-IRR.

C. *FedRater+ model analysis*

We examine the client-specific stateful parameter α_k (Line 19, Algorithm 1) and its relationship to c-IRR during model training. As shown in Fig. 1A, this correlation between α and c-IRR per rater for BRAVE increases across training rounds (we observed similar trends IEMOCAP and GoEmotions). After 20K rounds of training, the correlation between α and c-IRR was significant (permutation test $n = 10^5, p < 0.001$) with 0.30, 0.42 and 0.48 for BRAVE, IEMOCAP and GoEmotions respectively. It suggests that the client-specific parameter in FedRater+ can learn a proxy of "rater quality" in the distributed federated setting. Furthermore, as shown in Fig 1B, while α is positively correlated with c-IRR, it is negatively correlated with the number of samples available per client. The disassociation of α with respect to inter-rater agreements and rater sample sizes suggests that FedRater+ is able to capture aspects of "rater quality" that are complementary to merely how many samples a rater may have labeled.

VII. Conclusion

In this paper, we describe two parallel paradigms to model frequently occurring affective computing tasks in a federated learning (FL) setting: *user-as-client* and *rater-as-client*. We conducted a detailed empirical evaluation to study the use of FL to classify affective constructs from self-reports and perception ratings in audio, video and text datasets. First, where users are treated as clients, our results suggest that the performance of widely used (FedAvg) algorithm is comparable to that of centralized models, when there are a sufficient number of clients reporting updates to the server. In the *rater-as-client* setting, results suggest that our proposed algorithms FedRater and FedRater+ outperform FedAvg by 2% points average precision. A post-hoc analysis showed that the client-specific parameters learnt by the FedRater+ capture proxies of inter-rater agreement, enabling the server model in FL to behave as an "average rater" for predicting emotion perception labels.

VIII. Ethical Impact Statement

Developing machine learning models to classify affective constructs such as emotion is an open area and needs further research to understand the scope of their use. We note that these models do not try to infer the internal emotional state of individuals, but instead look at proxies such as facial or vocal expressions that may suggest one's expressed or perceived emotional state. In addition, the area of federated learning research and its application to model human-perception as in the case of affective computing is a new and evolving area of research. Our work does not currently take into account socio-cultural differences of the population studied in the federated setting and potential fairness issues that are typical to federated learning.

Authorized licensed use limited to: GOOGLE. Downloaded on April 01,2023 at 14:53:22 UTC from IEEE Xplore. Restrictions apply.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[3] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, H. Qi, D. Ramage, R. Raskar, M. Raykova, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[4] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[5] F. Granqvist, M. Seigel, R. van Dalen, Á. Cahill, S. Shum, and M. Paulik, "Improving on-device speaker verification using federated learning with privacy," *arXiv preprint arXiv:2008.02651*, 2020.

[6] 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)*. IEEE, 2021, pp. 3080–3084.

[7] S. Latif, S. Khalifa, R. Rana, and R. Jurdak, "Federated learning for speech emotion recognition applications," in *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2020, pp. 341–342.

[8] V. Tsouvalas, T. Ozcelebi, and N. Meratnia, "Privacy-preserving speech emotion recognition through semi-supervised federated learning," *arXiv preprint arXiv:2202.02611*, 2022.

[9] J. J. Sun, T. Liu, A. S. Cowen, F. Schroff, H. Adam, and G. Prasad, "Eev dataset: Predicting expressions evoked by diverse videos," *arXiv e-prints*, pp. arXiv–2001, 2020.

[10] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[11] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, "GoEmotions: A Dataset of Fine-Grained Emotions," in *58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[12] T.-M. H. Hsu, H. Qi, and M. Brown, "Federated visual classification with real-world data distribution," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer, 2020, pp. 76–92.

[13] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N project report, Stanford*, vol. 1, no. 12, p. 2009, 2009.

[14] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

[15] A. Cowen, G. Prasad, M. Tanaka, Y. Kamitani, V. Kirilyuk, K. Somandepalli, B. Jou, F. Schroff, A. Hartwig, J. A. Brooks *et al.*, "How emotion is experienced and expressed in multiple cultures: a large-scale experiment," *PsyArXiv*, 2021.

[16] C. O. Alm, "Subjective natural language problems: Motivations, applications, characterizations, and implications," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 107–112.

[17] S. Yang, H. Park, J. Byun, and C. Kim, "Robust federated learning with noisy labels," *arXiv preprint arXiv:2012.01700*, 2020.

[18] A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on cifar-10," *Unpublished manuscript*, vol. 40, no. 7, pp. 1–9, 2010.

[19] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2691–2699.

[20] C. Li, D. Niu, B. Jiang, X. Zuo, and J. Yang, "Meta-har: Federated representation learning for human activity recognition," in *Proceedings of the Web Conference 2021*, 2021, pp. 912–922.

[21] A. Metallinou and S. Narayanan, "Annotation and processing of continuous emotional attributes: Challenges and opportunities," in *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*. IEEE, 2013, pp. 1–8.

[22] M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," *Proceedings of the National Academy of Sciences*, vol. 110, no. 15, pp. 5802–5805, 2013. [Online]. Available: https://www.pnas.org/content/110/15/5802

[23] Y. Chen, X. Yang, X. Qin, H. Yu, P. Chan, and Z. Shen, *Dealing with Label Quality Disparity in Federated Learning*. Cham: Springer International Publishing, 2020, pp. 108–121.

[24] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds." *Journal of Machine Learning Research*, vol. 11, no. 4, 2010.

[25] K. Audhkhasi and S. Narayanan, "A globally-variant locally-constant model for fusion of labels from multiple diverse experts without using reference labels," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 4, pp. 769–783, 2012.

[26] R. Tanno, A. Saeedi, S. Sankaranarayanan, D. C. Alexander, and N. Silberman, "Learning from noisy labels by regularized estimation of annotator confusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 244–11 253.

[27] F. Eyben, F. Weninger, M. Woellmer, and B. Schuller, "the munich versatile and fast open-source audio feature extractor," *Proceedings ACM Multimedia (MM)*, pp. 1459–1462, 2018.

[28] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan *et al.*, "The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing," *IEEE transactions on affective computing*, vol. 7, no. 2, pp. 190–202, 2015.

[29] W. Jiang, Z. Wang, J. S. Jin, X. Han, and C. Li, "Speech emotion recognition with heterogeneous feature unification of deep neural network," *Sensors*, vol. 19, no. 12, p. 2730, 2019.

[30] R. Vemulapalli and A. Agarwala, "A compact embedding for facial expression similarity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5683–5692.

[31] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[33] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

Authorized licensed use limited to: GOOGLE. Downloaded on April 01,2023 at 14:53:22 UTC from IEEE Xplore. Restrictions apply.