ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10448318

FEDERATED LEARNING UNDER RESTRICTED USER AVAILABILITY

Periklis Theodoropoulos, Konstantinos E. Nikolakakis, Dionysis Kalogerias

Department of EE—Yale University

{periklis.theodoropoulos, konstantinos.nikolakakis, dionysis.kalogerias}@yale.edu

ABSTRACT

Federated Learning (FL) is a decentralized machine learning framework that enables collaborative model training while respecting data privacy. In various applications, non-uniform availability or participation of users is unavoidable due to an adverse or stochastic environment, the latter often being uncontrollable during learning. Here, we posit a generic user selection mechanism implementing a possibly randomized, stationary selection policy, suggestively termed as a Random Access Model (RAM). We propose a new formulation of the FL problem which effectively captures and mitigates limited participation of data originating from infrequent, or restricted users, at the presence of a RAM. By employing the Conditional Value-at-Risk (CVaR) over the (unknown) RAM distribution, we extend the expected loss FL objective to a risk-aware objective, enabling the design of an efficient training algorithm that is completely oblivious to the RAM, and with essentially identical complexity as FedAvg. Our experiments on synthetic and benchmark datasets show that the proposed approach achieves significantly improved performance as compared with standard FL, under a variety of setups.

Index Terms— Federated Learning, Conditional Value-at-Risk, Risk-Aware Learning, Stochastic Optimization, Random Access.

1. INTRODUCTION

Federated learning (FL) is a distributed learning framework allowing multiple users to train a global model collaboratively without sharing their local data [1]. In recent years, the classical Federated Averaging approach (FedAvg) has developed into an essential learning paradigm [2], following a certain basic workflow where each user locally updates its own model parameters and then periodically sends its updated parameters to a central server. The server then appropriately aggregates the received updates of the users and broadcasts to them the new global model parameters. This scheme is repeated until the global model converges [3].

Vanilla FedAvg faces the problem of instability, caused either due to non- indepedent and identically distributed (non-iid) data among users (known as ”client-drift”) [4–6], or due to non-uniform user availability, which happens as a result of several factors, such as network outages [7], battery drain [8], or user inactivity [9]. For instance, when a user device is unavailable, it cannot communicate with the server for a certain time interval. This leads to incomplete global model convergence, to a possible bias of the global model towards the data from the most available users, and less accurate performance of the global model to the data that the less available user have, especially in a strongly heterogeneous regime.

In prior works, several methods address the challenge of non-uniform client availability in FL. For the (server) aggregation step, a set of techniques focuses on re-weighting the model parameters of the users by adapting their weights dynamically. In particular, some works focus on the adaptation of the aggregation coefficients based on several criteria, such as completion of the local steps the users have done [9], or the temporal correlation and their low availability [10], or their conformity level [11], or their performance on previous steps [12]. Moreover, the non-uniform availability of users could be treated by dividing users into blocks based on their relative frequency, and applying a pluralistic aggregation step at each block has been proposed in [8]. Lastly, [13] suggests a different FL system that adopts a multiple-channel approach, following specifically the ALOHA protocol and adapting the access probability of users based on their local updates.

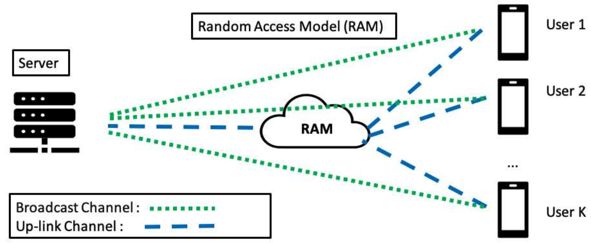


Fig. 1: Users send their updates to the RAM, which relays the data of only one user to the server. The server broadcasts the received local model to all users, and the process repeats.

A second group of techniques explores optimal user sampling strategies. Selection of users based on local characteristics, such as their local performance, level of importance, or irrelevance, is proposed in [14–16], respectively. Optimal user selection has also been explored by minimizing the variance of the numbers of times the server samples a client [17], or by learning a selection strategy for clients with intermittent availability [18]. Further, [19] uses stratified user sampling based on their data statistics to address system-induced bias under time-varying client availability. Lastly, a well-defined communication protocol where the server periodically selects user devices that meet appropriate criteria, is proposed in [20].

In this work, we deal with two key limitations, critical in practical scenarios. First, we consider a stochastic environment allowing only one (for simplicity) user to transmit data each time, with different and possibly highly biased selection probabilities (weights) for each user (restricted random user selection). Secondly, the server is completely agnostic to user selection probabilities, and cannot direct user participation. Under these conditions, the server communicates with the users through a noisy channel potentially adversarial to the learning procedure. We posit an intermediate provider, e.g., a multiplexer or switch, which we call the *Random Access Model (RAM)*, between users and the server. The RAM is responsible for relaying user model updates to the server by selecting a certain number of users to relay at each update (and communication) round.

The user selection criteria of the RAM remain unknown to both the server and users. Therefore, at least from the perspective of the server and users, the simplest approach to describe RAM user selection is by using a memoryless probabilistic model, i.e., the RAM spits out each user with some fixed probability, independently across communication rounds. In other words, the RAM acts as a station-

This work is supported by the NSF under grant CCF 2242215.

ary *erasure channel*, where at each round the updates of only one user survive, while the updates of the rest of the users are discarded, according to a fixed user selection distribution. That is, the RAM implements a possibly randomized user selection policy. Fig. 1 depicts this scheme, where the RAM allows only one user to relay its local model to the server each time. The server cannot intervene and works as a simple broadcaster.

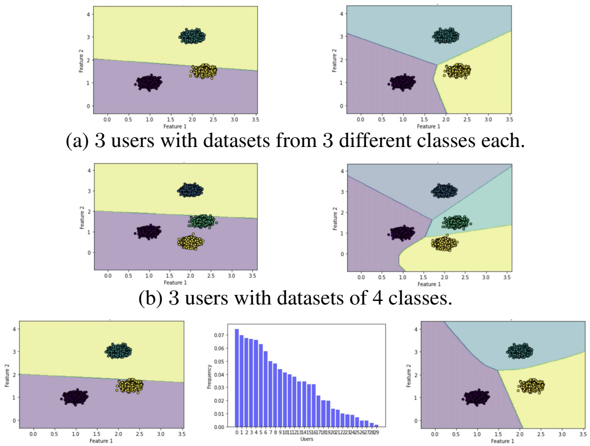
Under this setting, we propose a FL approach which is agnostic to user selection implemented by the RAM, by extending the expected empirical loss to a weighted risk-aware objective, using the *Conditional Value-at-Risk* (CVaR) over the (unknown) distribution dictating such selection of users. This approach is, to the best of our knowledge, new and results in a robust training algorithm which exploits the structure of FedAvg, with *at most* identical computational, iteration and communication complexity, at the expense of tuning two additional hyper-parameters. Our approach demonstrates notably superior performance compared with standard FL, which struggles to achieve accurate data classification when operating under limited access to user updates. Our experimental evaluations take place on both synthetic and standard benchmark datasets.

As a motivating example we consider a toy logistic regression problem on 2D synthetic data in Fig. 2. Each user trains on distinct patterns; this setup may resemble, e.g., sensors at different locations, each observing features corresponding to those patterns. Even for an easily verifiable setting with three or four classes, we observe that standard FL fails to classify data from less frequent users, while the proposed approach succeeds at finding decision boundaries that correctly classify the data from all users. All experiments were performed under the same training conditions (number of global rounds, step sizes, etc.). This example is elaborated in detail in Section 3.1.

1.1. Applicability of the RAM

The RAM provides an abstraction for capturing intricacies involved in the communication among network nodes in numerous networking applications. In the following, we discuss some relevant examples. More specifically, user participation with different relative frequencies naturally appears in the context of routing or switching. In this case, the RAM may embody a router or switch operating at the network layer, implementing some possibly randomized (steady-state) device selection policy, with the goal of routing numerous requests intelligently. The RAM may also model the role of the MAC sub-layer of a data-link network layer in managing access to a shared communication medium and potentially optimizing traffic prioritization for multiple users by, e.g., performing informed bandwidth allocation to different devices. Further, the RAM could describe potential user unavailability as an impact of network infrastructure at the physical layer. In this case, the RAM models the effects of varying traffic or interference patterns, resulting from allocating physical resources, such as power or frequency (carrier) directly.

In all those setups, highly biased and non-uniform user selections may result naturally under common circumstances. These include, for instance, service provider imbalanced user priorities due to certain contracts or monetary constraints, geographical constraints (e.g., communication in rural areas or in underwater applications), frequency band availability or scarcity in the physical layer, and opportunistic resource allocation policies. Additionally, user selection bias may be induced as a result of network outages and disruptions caused by environmental conditions, equipment failures, power outages, or even natural disasters, affecting user network access dramatically and thus causing non-uniformity in the availability of user data. Last but not least, user selection bias may result due to communication rate limitations caused by privacy concerns and censoring related to data from certain sources or of certain types.



(a) 3 users with datasets from 3 different classes each.

(b) 3 users with datasets of 4 classes.

(c) 30 users with datasets of 3 classes.

Fig. 2: Efficacy of our approach on a synthetic logistic regression example (Left: FL | Right: Proposed | Center: RAM Distribution).

The RAM provides a convenient abstraction for modeling such non-uniform user selection schemes, which introduce unavoidable data scarcity and result in rare user participation in a FL setup.

1.2. Comparison with previous works

Due to rare user participation enforced by the RAM, together with its unknown structure, existing techniques (e.g., user re-weighting [9–12] and optimal sampling policies [14–19]) are inapplicable, as they require server-user interaction. In contrast, the proposed algorithm allows the server to manage user selection bias by optimizing efficiently under data scarcity. Our algorithm can in fact systematically manage non-uniform user participation, by adaptively focusing more on less frequent users. This is possible by exploiting native properties of the CVaR. The proposed technical approach also shares some similarities with [21, 22]. However, our formulation is fully interpretable (i.e., all hyperparameters admit a fully specified operational meaning), while the resulting algorithm operates fully within the framework of FedAvg; no heuristics or customized algorithm design is necessary. Lastly, our approach resembles online federated learning [23] but differs in that users can only access the server when permitted by the RAM, even after completing local updates.

2. PROBLEM SETUP

We consider a multi-class classification setup, with features $x \in \mathcal{X} \subseteq \Re^d$ and labels $y \in \mathcal{Y} = \{1, 2, \cdots, C\}$, where C is the total number of classes (patterns). We also consider a federated learning setting, with users $i \in [1, \ldots, K]$ selected by the RAM from a fixed distribution $\mathcal{D}$, with probabilities ρ_i, such that $\sum_{i=1}^{K} \rho_i = 1$. Further, we assume that each user i, has a private dataset $D_i(x, y) := D_i(\xi)$, with N_i number of data. In the context of FL, the server updates the users via an aggregation step. Let us also define the family $\tilde{\Theta} := \{\phi : \mathcal{X} \times \Theta \rightarrow \Re^C\}$ of parameterized predictors with parameter $\theta \in \Theta \subseteq \Re^Q$. Each user tries to minimize its local loss function $l_i : \Re^C \times \mathcal{Y} \rightarrow \Re_+$.

In standard FL (and under our setting), the goal of the server would be to find optimal global parameters θ solving the problem

$$\inf_{\theta \in \Theta} \mathbb{E}_{I \sim \mathcal{D}} \Big[\mathbb{E}_{\xi \sim D_I} \big[l_I(\phi(\xi; \theta)) \big] \Big], \qquad (1)$$

whose empirical version reads as

$$\inf_{\theta\in\Theta}\left\{G_{\text{FL}}(\theta) := \sum_{i=1}^{K}\rho_i\frac{1}{N_i}\sum_{j=1}^{N_i} l_i(\phi(\xi_j;\theta)) = \sum_{i=1}^{K}\rho_i f_i(\theta)\right\}. \tag{2}$$

Recall that the RAM applies some user selection policy which is *both unknown and untouchable* from the side of the server. Therefore, any training algorithm should be agnostic to the RAM distribution, and it should work for any RAM *and* dataset distributions. If the RAM distribution is highly skewed, then problem (2) faces the issue of non-uniform and rare user participation. In such a case the data are distributed with strong heterogeneity among users, data starvation exists, and accurate classification by solving (2) is generally hopeless, as we demonstrate in Fig. 2(left).

3. PROPOSED APPROACH

To guarantee efficient classification and simultaneously mitigate the effect of data starvation, we propose a *risk-aware objective* that combines the *Conditional Value-at-Risk* (CVaR) on the RAM distribution with the empirical risk-neutral objective (2). The CVaR of a random variable Z at confidence level $\alpha \in (0,1]$ is defined as [24]

$$\text{CVaR}^{\alpha}\big[Z\big] := \inf_{t\in\Re}\left\{t + \frac{1}{\alpha}\mathbb{E}\big[(Z-t)_+\big]\right\}, \tag{3}$$

for which it is true that $\text{CVaR}^{\alpha}\big[Z\big] = \mathbb{E}[Z]$ for $\alpha = 1$, rising up to ess sup$Z$, as $\alpha \to 0$. Then, for $\gamma \in (0,1)$, our proposed optimization problem is

$$\inf_{\theta\in\Theta}\left\{G^{\text{R}}_{\text{FL}}(\theta) := (1-\gamma)\text{CVaR}^{\alpha}_{I\sim\mathcal{D}}\big[f_I(\theta)\big] + \gamma\mathbb{E}_{I\sim\mathcal{D}}\big[f_I(\theta)\big]\right\}, \tag{4}$$

and using definition (3) for $Z = f_I(\theta)$, we have

$$\inf_{\theta\in\Theta}(1-\gamma)\inf_{t\in\Re}\left\{t+\frac{1}{\alpha}\sum_{i=1}^{K}\rho_i\big[f_i(\theta)-t\big]_+\right\} + \gamma\sum_{i=1}^{K}\rho_i f_i(\theta). \tag{5}$$

The objective in the preceding problem may be further simplified as

$$\inf_{(\theta,t)}\sum_{i=1}^{K}\rho_i\left\{G_i(\theta,t):=(1-\gamma)\Big[t+\frac{1}{\alpha}\big[f_i(\theta)-t\big]_+\Big]+\gamma f_i(\theta)\right\}. \tag{6}$$

The CVaR measures expected losses restricted to the upper tail of the distribution of the random variable Z [25]. Thus, by tuning the parameters $\gamma \in [0,1]$ and $\alpha \in (0,1]$, we tune the objective in (4) to boost the learning procedure on data points that come from rare user participation events, and essentially enforce learning under data starvation with a few shots only. Equivalently, a training algorithm based on (4) learns how to reject samples from frequent users since $\text{CVaR}^{\alpha}[\cdot]$ is robust to the uncertainty of the environment [24] .

Problem (6) leads us to devise Algorithm 1 for tackling (4). Algorithm 1 is an extension of FedAvg, and essentially an instance of FedAvg on the proposed risk-aware problem (6). In each round, the server receives the parameters (θ_i^n, t_i^n) of a certain user i (chosen iid by the RAM and not by the server) and broadcasts those as global parameters $(\theta^n_{\text{global}}, t^n_{\text{global}})$ to all users. Then, each user locally applies gradient descent steps on its private dataset, updating its parameters θ_i^{n+1} and t_i^{n+1}.

We again note that the server is agnostic to RAM user selection. So, the proposed Algorithm 1 asks the users to tackle a more general problem than in the standard risk-neutral case (cf. (1)), to solve locally. Indeed, Algorithm 1 asks the users to optimize the risk-aware objective of (4) –through that of (6)–, given a desired CVaR confidence level a and a trade-off parameter γ. When $\alpha = 1$, (4) is reduced to the standard FL objective (1), and Algorithm 1 reduces to standard FedAvg.

Algorithm 1 FED-CVaR-AVG

```
Initialize $\theta_i^1 = \theta, t_i^1 = t$, for all $i$. Set $K, T, H, \gamma, \alpha$.
for each global round $n = 1, \dots, T$ do
    Server broadcasts the user selected by the RAM:
        $(\theta^n_{\text{global}}, t^n_{\text{global}}) \leftarrow \mathbf{RAM}(\{(\theta_i^n, t_i^n)\}_{i=1}^K)$
    for all users $i \in [K]$ in parallel do
        $\mathcal{B} \leftarrow$ split each $D_i$ into batches of size $B_i$
        $(\theta_i, t_i) \leftarrow (\theta^n_{\text{global}}, t^n_{\text{global}})$
        for local epoch $h = 1, \dots, H$ do
            for batch $b \in \mathcal{B}$ do
                $\begin{bmatrix}\theta_i \\ t_i\end{bmatrix} \leftarrow \begin{bmatrix}\theta_i - \eta_\theta\nabla_\theta G_i(b;\theta_i,t_i) \\ t_i - \eta_t\nabla_t G_i(b;\theta_i,t_i)\end{bmatrix}$
            end for
        end for
        Forward to RAM: $(\theta_i^{n+1}, t_i^{n+1}) \leftarrow (\theta_i, t_i)$
    end for
end for
```

3.1. A Motivating Example

We now present a simple example to illustrate the differences in behavior of the standard (risk-neutral) FL objective in (1) and the proposed risk-aware objective in (4). Suppose that a dataset is distributed among $K = 3$ users, with each user training for 1 pattern. Let us also assume that the RAM selects the users with probabilities $\rho_1 > \rho_2 \gg \rho_3$, which means that the RAM allows users 1, and 2, to communicate more often with the server than the user 3.

As usual, the classical FedAvg [2] approach will try to minimize the objective function

$$G_{\text{FL}}(\theta) = \sum_{i=1}^{K=3}\rho_i f_i(\theta) = \rho_1 f_1(\theta) + \rho_2 f_2(\theta) + \rho_3 f_3(\theta), \tag{7}$$

where the weights ρ_1, ρ_2 and ρ_3 are unknown, but implicitly supplied by the RAM.

On the other hand, for a sufficiently small and strictly positive choice of the hyper-parameter (the CVaR level) α, it can be easily shown (although not entirely trivially) that the positive part of the risk-aware objective in (4) is activated only for the upper α-quantile of the empirical losses $f_1(\theta), f_2(\theta), f_3(\theta)$ on the random variable I (for each fixed θ). This yields the *weighted user-robust loss*

$$G^{\text{R}}_{\text{FL}}(\theta) = (1-\gamma)\max\{f_1(\theta), f_2(\theta), f_3(\theta)\} + \gamma G_{\text{FL}}(\theta), \tag{8}$$

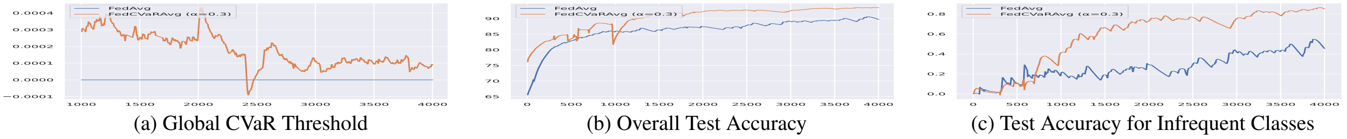
for every sufficiently small trade-off parameter $\gamma \in [0,1]$. We observe that the risk-aware objective (8) focuses on the worst user loss regardless of the corresponding probability of it being selected by the RAM, with relative proportion $1-\gamma$.

Therefore, in a region of the space where, e.g., f_3 is larger, which is expected to happen due to rare sampling by the RAM, the risk-aware objective (8) will steer θ towards regions of Θ that equalize (i.e., reduce) the values of the local training loss f_3, relative to f_1 and f_2. In other words, the objective (8) induces *user equity* in FL, which is initially hindered by the presence of the RAM.

It is worth-noting that while (8) is an operationally desirable objective, it is practically impossible for the server to infer which of the three losses is largest, since the RAM prevents the server from controlling user participation in learning. Additionally, (8) generally results in not well-behaved and possibly nonsmooth FL problems. In our approach, these challenges are effectively addressed by replacing the risk-aware problem (4) by its equivalent version (6), which is well-behaved and does not require access to unavailable information.

	$\alpha = 1.0$			$\alpha = 0.3$			$\alpha = 0.2$			$\alpha = 0.1$		
	Overall.	**pattern 1.**	**pattern 2.**	**Overall.**	**pattern 1.**	**pattern 2.**	**Overall.**	**pattern 1.**	**pattern 2.**	**Overall.**	**pattern 1.**	**pattern 2.**
$\gamma = 0.0$	85.147 ± 0.546	85.473 ± 1.548	61.041 ± 3.554	86.157 ± 0.559	88.323 ± 1.140	67.164 ± 3.344	86.453 ± 0.592	$\mathbf{88.736} \pm 1.472$	68.926 ± 2.716	86.089 ± 0.923	87.536 ± 2.449	70.601 ± 3.523
$\gamma = 0.1$	84.895 ± 0.605	84.583 ± 1.597	59.649 ± 3.795	86.276 ± 0.519	88.353 ± 1.058	66.676 ± 2.883	86.432 ± 0.653	88.619 ± 1.674	68.801 ± 2.835	$\mathbf{86.546} \pm 0.540$	88.681 ± 1.454	$\mathbf{71.681} \pm 2.419$
$\gamma = 0.2$	84.783 ± 0.474	84.533 ± 1.294	59.223 ± 2.580	86.174 ± 0.552	88.029 ± 1.572	66.372 ± 3.116	86.110 ± 0.573	88.061 ± 1.663	67.659 ± 3.156	86.132 ± 0.860	88.093 ± 1.990	69.698 ± 4.351
$\gamma = 0.3$	84.983 ± 0.419	84.969 ± 1.042	60.330 ± 2.104	86.036 ± 0.483	87.611 ± 1.316	65.958 ± 2.492	86.263 ± 0.517	88.300 ± 1.772	68.076 ± 3.054	86.197 ± 0.533	87.776 ± 2.026	69.883 ± 2.674
$\gamma = 1.0$	84.915 ± 0.497	84.514 ± 1.620	60.252 ± 3.074	84.822 ± 0.630	84.481 ± 2.003	59.008 ± 3.138	84.916 ± 0.454	85.459 ± 2.870	60.805 ± 2.399	84.956 ± 0.462	85.561 ± 1.443	59.839 ± 3.168

Table 1: **FashionMnist:** The 10% of less available users (that means 3 out of 30 users with sampling probabilities $0.0107, 0.0078, 0.0053$, respectively) carry 2 patterns exclusively. The columns represent the overall testing accuracy, and the testing accuracy, of the global model, at the patterns that belong to the less available users, respectively. The objective becomes risk-neutral when $\alpha = 1.0$, or $\gamma = 1.0$.



(a) Global CVaR Threshold

(b) Overall Test Accuracy

(c) Test Accuracy for Infrequent Classes

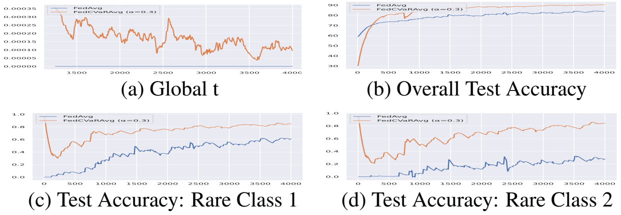
Fig. 3: **MNIST**: For $K = 30$ users, with the 3 of less often users have exclusively 1 of the patterns (x-axis is global round index).

The sharp distinction between (7) and (8) is demonstrated in Fig. 2, for a simple logistic regression setup on 2D synthetic data. Specifically, in Fig. 2a, each of $K = 3$ users receives one pattern, while in 2b, an extra pattern is assigned to the least frequent user 3. User selection probabilities are $\rho_1 = 0.5, \rho_2 = 0.4$, and, $\rho_3 = 0.1$. We observe that the standard FL ($\alpha = 1$) fails to classify correctly the pattern/s from the least frequent user 3. However, the proposed approach which solves (6) (for $\alpha = 0.1$) generates decision boundaries that correctly classify all patterns in both cases. Similarly, in Fig. 2c, we scale up to $K = 30$ users, with the 6 least frequent users training on the yellow dataset. The user selection probabilities are shown in 2c(center). Once again, the risk-aware approach ($\alpha = 0.1$) successfully generates a decision boundary that classifies all patterns. Across all examples shown in figure 2, the trade-off γ remains constant at 0.1 and all experiments were performed fairly under exactly the same choices regarding algorithm hyperparameters, number of epochs (all models are "trained to plateau"), etc.

4. EXPERIMENTAL RESULTS

We now evaluate the proposed Algorithm 1 on the **Mnist** (Figs. 3 and 4) and **FashionMnist** (Table 1) benchmarks, each comprised of 60,000 training samples with 10 distinct patterns. Data are distributed among K users in a heterogeneous way. We split the data, reserving $M\%$ of the most frequent users for $r\%$ of available patterns, while the remaining $(100 - r)\%$ of the data is uniformly distributed among the least frequent users, comprising the remaining $(100 - M)\%$. The total number of global rounds is the same for both FedAvg and the proposed algorithm, and set to 4000 and 6000 for the **Mnist** and **FashionMnist** datasets, respectively. Code for all experiments is available at [26], together with additional and more detailed figures (presented briefly here due to space limitations).

For the **Mnist** dataset, we present results for two experiments with $K = 30$ users. For the first experiment in Fig. 3, we set $M = 90\%$, and $r = 90\%$, and for the second experiment in Fig. 4, we set $M = 90\%$, and $r = 80\%$. We also choose $\alpha = 0.3$ and $\gamma = 0.3$. For both FedAvg and Algorithm 1 we use a neural network with two fully-connected hidden layers, with number of neurons $(128, 128)$ [27]. Stepsizes are set constant as $\eta_\theta = 10^{-3}$, $\eta_t = 10^{-4}$, and each user conducts 10 local epochs. We report smoothed graphs for clarity. In both Figs. 3 and 4, we readily observe that Algorithm 1 achieves both better overall performance and better performance at the patterns that are locally trained by the least frequent users. We can observe performance improvement to over 80%, from around 50% (Fig. 3c), 60% (Fig. 4c) and 30% (Fig. 4d), respectively.



(a) Global t

(b) Overall Test Accuracy

(c) Test Accuracy: Rare Class 1

(d) Test Accuracy: Rare Class 2

Fig. 4: **MNIST**: For $K = 30$ users, with 3 of less often users have exclusively the 2 of the patterns.

For the **FashionMnist** dataset, Table 1 provides a more detailed range of experiments. We set $K = 30$, $M = 90\%$ and $r = 80\%$, and use a CNN similar as in [2] with two 5×5 convolutional layers (with 6 and 16 channels, respectively, each followed with 2×2 max pooling) and two fully connected layers with 120 and 84 neurons. Stepsizes are set as $\eta_\theta = 10^{-2}$ and $\eta_t = 5 \times 10^{-4}$. Each user runs 20 local epochs. The setting remains the same for all α and γ.

Table 1 shows the overall global performance as well as that for the patterns belonging to the least frequent users, for a variety of pairs (γ, α). The blue area of Table 1 corresponds to cases where problem (4) is essentially reduced to standard FL. On the other hand, in the orange area the objective of (4) becomes most risk-sensitive. We observe that the accuracy at pattern 2 has been improved from approximately 60% for standard FL to more than 71%, for $\alpha = 0.1$ and $\gamma = 0.1$. Moreover, performance on pattern 1 is also improved as compared with FedAvg. Further, it is worth mentioning that pattern 2 is most difficult to learn even for the standard FL, compared with pattern 1. As expected, overall performance improves when $\alpha \to 0$, with the case $(\gamma = 0.1, \alpha = 0.1)$ performing the best.

5. CONCLUSION

In this work, we studied FL under an unknown random access model (RAM) describing biased, non-uniform, skewed and/or restricted random user participation. Departing from the standard expected loss model, we proposed a new risk-aware objective constructed by taking the CVaR over the RAM distribution, resulting in an efficient training algorithm which is oblivious to the RAM, but at the same time addresses limited participation of infrequent users. Through experimental evaluation on 2D synthetic, Mnist, and FashionMnist datasets, we have demonstrated that the proposed risk-aware approach brings substantial potential performance gains over standard FL relying on FedAvg which, to the best of our knowledge, is currently a state-of-the-art method for handling FL problems with no server intervention on user participation.

6. REFERENCES

[1] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar, “Federated multi-task learning,” *Advances in neural information processing systems*, vol. 30, 2017.

[2] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

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

[4] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh, “Scaffold: Stochastic controlled averaging for federated learning,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

[5] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama, “Federated learning based on dynamic regularization,” *arXiv preprint arXiv:2111.04263*, 2021.

[6] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith, “Federated optimization in heterogeneous networks,” *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[7] Jianyu Wang et al., “A field guide to federated optimization,” *arXiv preprint arXiv:2107.06917*, 2021.

[8] Hubert Eichner, Tomer Koren, Brendan McMahan, Nathan Srebro, and Kunal Talwar, “Semi-cyclic stochastic gradient descent,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 1764–1773.

[9] Yichen Ruan, Xiaoxi Zhang, Shu-Che Liang, and Carlee Joe-Wong, “Towards flexible device participation in federated learning,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 3403–3411.

[10] Angelo Rodio, Francescomaria Faticanti, Othmane Marfoq, Giovanni Neglia, and Emilio Leonardi, “Federated learning under heterogeneous and correlated client availability,” *arXiv preprint arXiv:2301.04632*, 2023.

[11] Yassine Laguel, Krishna Pillutla, Jérôme Malick, and Zaid Harchaoui, “A superquantile approach to federated learning with heterogeneous devices,” in *2021 55th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2021, pp. 1–6.

[12] Zhiyuan Zhao and Gauri Joshi, “A dynamic reweighting strategy for fair federated learning,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8772–8776.

[13] Rafael Valente da Silva, Jinho Choi, Jihong Park, Glauber Brante, and Richard Demo Souza, “Multichannel aloha optimization for federated learning with multiple models,” *IEEE Wireless Communications Letters*, vol. 11, no. 10, pp. 2180–2184, 2022.

[14] Sumit Rai, Arti Kumari, and Dilip K Prasad, “Client selection in federated learning under imperfections in environment,” *AI*, vol. 3, no. 1, pp. 124–145, 2022.

[15] Wenlin Chen, Samuel Horvath, and Peter Richtarik, “Optimal client sampling for federated learning,” *arXiv preprint arXiv:2010.13723*, 2020.

[16] Yae Jee Cho, Jianyu Wang, and Gauri Joshi, “Client selection in federated learning: Convergence analysis and power-of-choice selection strategies,” *arXiv preprint arXiv:2010.01243*, 2020.

[17] Zheng Wang, Xiaoliang Fan, Jianzhong Qi, Haibing Jin, Peizhen Yang, Siqi Shen, and Cheng Wang, “Fedgs: Federated graph-based sampling with arbitrary client availability,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023, vol. 37, pp. 10271–10278.

[18] Mónica Ribero, Haris Vikalo, and Gustavo De Veciana, “Federated learning under intermittent client availability and time-varying communication constraints,” *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[19] Ming Tang and Vincent WS Wong, “Tackling system induced bias in federated learning: Stratification and convergence analysis,” in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[20] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al., “Towards federated learning at scale: System design,” *Proceedings of machine learning and systems*, vol. 1, pp. 374–388, 2019.

[21] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith, “Fair resource allocation in federated learning,” *arXiv preprint arXiv:1905.10497*, 2019.

[22] Guojun Zhang, Saber Malekmohammadi, Xi Chen, and Yaoliang Yu, “Proportional fairness in federated learning,” *arXiv preprint arXiv:2202.01666*, 2022.

[23] Aritra Mitra, Hamed Hassani, and George J Pappas, “Online federated learning,” in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 4083–4090.

[24] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*, MOS-SIAM Series on Optimization. SIAM & Mathematical Optimization Society, Philadelphia, 2014.

[25] Dionysios S Kalogerias, “Noisy linear convergence of stochastic gradient descent for cv@ r statistical learning under Polyak-Lojasiewicz conditions,” *arXiv preprint arXiv:2012.07785*, 2020.

[26] Periklis Theodoropoulos, “Risk-aware federated learning: Simulations and experiments,” https://github.com/PeriklisTheodoropoulos/risk-aware-FL, 2023.

[27] Zebang Shen, Juan Cervino, Hamed Hassani, and Alejandro Ribeiro, “An agnostic approach to federated learning with class imbalance,” in *International Conference on Learning Representations*, 2021.