Communication efficient model-aware federated learning for visual crowd counting and density estimation in smart cities

A. Armacki, N. Milosevic, D. Bajovic, S. Kar, D. Jakovetic, A. Bakhtiarnia, L. Esterle,
A. Muscat, and T. Festi

***Abstract*—Federated learning (FL) is an attractive paradigm where a number of users can improve their local models via sharing trained models or model increments with a central server, while the users' data is kept private. However, when model sizes are huge, FL incurs a significant communication overhead. In such scenarios, strategies that perform user sampling, so that only informative users communicate their models to the server, are desired. In this paper, we make several contributions on user sampling in FL. On the theoretical side, we consider a general framework that exhibits user heterogeneity across several dimensions: activation probabilities, gradient noise variance, number of updates per epoch, and communication channel quality. In this setting, we derive convergence rate of the FedAvg method. The rate explicitly characterizes the effects of heterogeneity and enables us to derive optimal user sampling probabilities in an offline setting, when the sampling probabilities are pre-computed. We then show how these derived probabilities naturally connect with existing optimized sampling strategies in an adaptive-online setting. On the practical side, we study visual crowd counting (VCC) as a representative deep learning application with huge-sized models. We provide an implementation of the FL system over real-world data across three pilot sites – Valetta, Trento and Novi Sad. The evaluation results demonstrate significant model accuracy benefits through employing FL over the multiple cities, and significant communication savings via non-uniform user sampling strategies.**

***Index Terms*—Federated learning, Optimized client selection, Visual crowd counting, Communication efficient protocol.**

I. Introduction

Federated learning (FL) refers to the concept where a number of users collaboratively learn a global machine learning model with the assistance of a global coordinator (server), e.g., [1]. During the FL process, users do not share their raw data but only share locally trained models or model increments with the server, hence making FL attractive from data protection and privacy points of view in various applications [1], e.g., visual crowd counting (VCC) in smart cities [2].

In this paper, in pursuit of communication-efficiency [3], the strategy of interest is optimized user sampling, i.e., selecting a subset of appropriately defined most informative users at each training epoch. This is in contrast with the standard uniform sampling strategy in FL that selects S out of N users uniformly at random; see, e.g., [4]–[6]. In more detail, we consider a general non-uniform sampling framework for FL in the presence of various sources of system heterogeneity. Therein, each user i is active, i.e., transmits its model increment, with a user-dependent probability q_i. The users perform the federated averaging (FedAvg) training method [7], where each user at each epoch makes a user-dependent number K_i of local stochastic gradient (SGD) steps. The user-dependent K_i's model heterogeneous users' capabilities in terms of computational and storage power, processor speed, etc, e.g., [8]. In addition, the noise variances introduced with users' local SGD steps are user-dependent, hence modeling different data quality or different batch sizes across users. Finally, we allow that the user-server links may be unreliable, so that transmission of an active user is received at the server with a user-dependent probability k_i. The k_i's here account for communication channel imperfections, such as finite user transmit power, packet droputs, etc.

There have been several recent works that consider non-uniform user sampling in FL and propose optimized user sampling strategies, e.g., [9]–[11]. A user sampling strategy can be considered in an offline setting, where user activation probabilities q_i's are determined beforehand. On the other hand, in an online setting, user activation probabilities are calculated adaptively at each training epoch based on the FL algorithm progress.

In this paper, we make several theoretical and practical contributions towards better understanding of how users should be sampled for communication-efficient FL. First, on the theoretical side, we consider a general heterogeneous offline user sampling framework, as described above. For this framework, we derive convergence rate of FedAvg assuming either strongly convex or convex user losses. We then explicitly quantify the achieved rate with respect to various sources of

A. Armacki and S. Kar are with Carnegie Mellon University, Pittsburgh, PA, USA. N. Milosevic and D. Jakovetic are with Faculty of Sciences, University of Novi Sad, Serbia. D. Bajovic is with Faculty of Technical Sciences, University of Novi Sad, Serbia. A. Bakhtiarnia and L. Esterle are with Aarhus University, Denmark. A. Muscat is with Greenroads, Malta, and with University of Malta. T. Festi is with Comune di Trento, Italy. Authors' emails: aarmacki@andrew.cmu.edu, nmilosev@dmi.uns.ac.rs, dbajovic@uns.ac.rs, soummyak@andrew.cmu.edu, dusan.jakovetic@dmi.uns.ac.rs, [arianbakh,lukas.esterle]@ece.au.dk, adrian@greenroadsmalta.com, thomas.festi@comune.trento.it. The work of NM, DB, DJ, AB, LE, AM and TF was supported in part by the European Union's Horizon 2020 Research and Innovation program under grant agreement No 957337. The paper reflects only the view of the authors and the Commission is not responsible for any use that may be made of the information it contains. The work of NM, DB and DJ was also supported in part by the Serbian Ministry of Education, Science and Technological Development.

system heterogeneity. Next, optimal user activation probabilities q_i's are derived that maximize the convergence rate subject to a given per-epoch communication budget and user-server link capacities, i.e., probabilities k_i's. These results generalize existing bounds [9] to the case of heterogeneous local user variances, and, more importantly, to heterogeneous numbers of local SGD updates K_i's. This reveals several useful insights regarding practical user sampling strategies in FL. First, a user sampling probability should be inversely proportional to its number of local updates K_i; intuitively, performing more local work leads to better variance reduction and more informative updates, allowing users to communicate with the FL server less frequently. On the other hand, interestingly, the user activation probability should be proportional to its local variance, i.e., higher variance users should be polled more frequently. This is explained intuitively by the fact that users with more inherent local noise should be active more frequently in order to filter out the noise, in view of the underlying law of large numbers phenomenon. For the online (adaptive) client sampling setting, we describe how the results we derive are in accordance with the optimal adaptive user selection strategy in [9].

Second, on the implementation and application side, we develop a FL system for VCC in smart cities, with a real-world system deployment over three cities, Valetta, Trento, and Novi Sad. VCC is an important task with numerous applications in smart cities, such as people density estimation in public squares, pedestrian counting in traffic areas, and monitoring large-scale events by unmanned aerial vehicles (UAVs). The deployed FL system for VCC supports heterogeneous client sampling. Specifically, we employ the optimal adaptive (online) client sampling strategy from [9]. Extensive numerical experiments demonstrate that, for the training sites (cities) that lack sufficient data beforehand, FL leads to significant accuracy improvements of VCC. Moreover, the results demonstrate significant communication savings when non-uniform, optimized user sampling is performed.

II. Problem setup

A. Federated learning model

We consider a FL system with N users that wish to collaboratively minimize the following function:

$$f(w) = \sum_{i=1}^{N} p_i f_i(w). \qquad (1)$$

Here, quantity $0 < p_i < 1$, $\sum_{k=1}^{N} p_i = 1$, and $f_i : \mathbb{R}^d \mapsto \mathbb{R}$ is the local user i's cost function, e.g., the empirical or population loss of user i with respect to its local data (local data distribution).

In order to solve (1) in the FL setting, users and the server perform the FedAvg algorithm with user-dependent number of local SGD updates and a heterogeneous probabilistic user activation scheme. We denote by $t = 0, 1, ...$ the outer iteration (epoch) counter, and by $j = 0, 1, ...$ the inner iteration index that counts the number of local SGD updates at each user. To be more precise, we denote by $w_{t,j}^i \in \mathbb{R}^d$ the user i's estimate of the solution to (1) at outer iteration (epoch) t and inner SGD iteration j, $j = 1, ..., K_i$. Similarly, denote by $w_t \in \mathbb{R}^d$ the server's solution estimate at epoch t, $t = 0, 1, ...$ Further, denote by $g_{t,j}^i$ user i's noisy estimate of gradient $\nabla f_i(w_{t,j}^i)$ at point $w_{t,j}^i$, and let $g_t^i = \frac{1}{K_i}\sum_{j=0}^{K_i-1} g_{t,j}^i$. Next, let X_t^i be a Bernoulli random variable that encodes the information whether the server receives the locally updated model w_{t,K_i}^i from user i at epoch t. We let $X_t^i \sim Bernoulli(q_i)$, with parameter q_i. In other words, q_i is the probability that the server successfully receives the model w_{t,K_i}^i at the end of the inner iteration process at epoch t from user i. Quantity X_t^i in our setting models two effects: user i's random activation (user decides to be active or not); and message reception failure if the model was transmitted but was not received by the server[1] [12]. The algorithm works as follows. For all $i = 1, ..., N$, at each epoch t, each user $i = 1, ..., N$ performs the following update:

$$w_{t,j+1}^i = w_{t,j}^i - \alpha\, g_{t,j}^i, \qquad (2)$$

for $j = 0, ..., K-1$. Here, $\alpha > 0$ is the step-size, and $w_{t,0}^i = w_t$. Then, the server aggregates model increments $g_i^t = \frac{w_{t,K_i}^i - w_{t,0}^i}{\alpha K_i}$ from all users i for which $X_t^i = 1$, and computes

$$w_{t+1} = w_t - \alpha \sum_{i=1}^{N} p_i \frac{X_{t+1}^i}{q_i} g_t^i. \qquad (3)$$

Note that we implicitly assume that the server knows quantities q_i's, $i = 1, ..., N$. The division of each user's contributing term by q_i in (3) enables elimination (on average) of the non-uniform sampling bias, similarly to how dividing the local aggregate updates by $\frac{1}{K_i}$ ensures objective consistency, e.g., [8]. In other words, there holds: $\mathbb{E}[w_{t+1} \,|\, \mathcal{F}_t] = w_t - \alpha \sum_{i=1}^{N} p_i g_t^i$. Here, $\mathbb{E}[\cdot|\cdot]$ denotes conditional expectation, $\mathcal{F}_t$ is the history of the algorithm, including all the algorithmic steps history prior to the generation of the instances of the variables X_t^i, $i = 1, ..., N$.

B. Visual crowd counting

VCC corresponds to counting the total number of people that are present in a given scene. The input data X to the model is a high-resolution RGB colour space image of a scene containing people, and the output Y that this model provides is a density map, which is a single-channel image that specifies the density of the crowd at each pixel of the input image (similar to image masks in image segmentation tasks). The values in this density map can be summed to get the total count in the form of a single number. The annotations used for training this model are in the form of head annotations, where the location of the center of each person's head is specified. The model utilizes the SASNet neural network architecture [13]. In other words, we employ a model-aware approach, where each FL user pre-assumes the

[1] More precisely, we can let $X_t^i = Z_t^i Y_t^i$, where Z_t^i and Y_t^i are both Bernoulli random variables, Z_t^i indicating whether agent i was selected to send its update during communication round t, while Y_t^i indicating if the communication was successful or not.

specific SASNet architecture that is in prior works proven to be effective in VCC in a single-user setting [13]. Then, the weights of SASNet are learned via FL, harnessing data from all FL users. The FL users here correspond to $N = 3$ sites in 3 different cities (Valetta, Trento, Novi Sad). In view of (1), the unknown model $w \in \mathbb{R}^d$ thus represents the unknown weights of the SASNet neural network that maps inputs X to outputs Y, while function f_i represents the training loss with respect to the input-output data acquired at FL user i. We refer to [13] for further SASNet model and loss function details.

III. Convergence rate and optimal activation

A. Convergence rate

We next present convergence rate results for the FL algorithm (2)–(3) under the heterogeneous setting described in Section II. For a cleaner presentation, we let $p_i = 1/N$, for all i (see (1)), while the results generalize to arbitrary p_i's. We make the following assumptions.

Assumption 1. *Each f_i is L-smooth and μ-strongly convex, i.e., the following inequalities are satisfied for all $x, y \in \mathbb{R}^d$, for some constants $0 < \mu \leq L$: $\big\|\nabla f_i(x) - \nabla f_i(y)\big\| \leq L\|x - y\|$, and $\big\langle \nabla f_i(x) - \nabla f_i(y), x - y\big\rangle \geq \mu\|x - y\|^2$, where $\|\cdot\|$ denotes the Euclidean norm.*

Assumption 2. *Each $g^i_{t,j}$ is an unbiased estimate of the true gradient, and the variance is bounded, i.e.:*

$$\mathbb{E}\big[g^i_{t,j} \big| w^i_{t,j-1}\big] = \nabla f_i(w^i_{t,j}),$$
$$\mathbb{E}\Big[\big\|g^i_{t,j} - \nabla f_i(w^i_{t,j})\big\|^2 \big| w^i_{t,j-1}\Big] \leq \sigma_i^2,$$

for all $j = 1, \ldots, K_i$, for some constant $\sigma_i^2 > 0$, $i = 1, ..., N$.

Assumption 3. *The indicator variables X^i_t are independent across time and across users.*

For future reference, we introduce the following quantities: the maximal number of local updates $\overline{K} = \max_{i\in[N]} K_i$ (here, $[N] = \{1, 2, ..., N\}$); the local communication probability parameter $\overline{q}_i = \frac{1-q_i}{q_i}$; the minimal communication probability parameter $q_{\min} = \min_{i\in[N]} q_i$. We also denote by w^* a solution to (1).

We have the following result. The proof is provided in an extended version of this paper [14].

Theorem 1. *Let assumptions 1-3 hold. For the step-size satisfying $\alpha \leq \min\left\{\frac{q_{\min}}{64L}, \frac{1}{10L\overline{K}}\right\}$, we have the following convergence guarantees:*
1) *For f general convex ($\mu = 0$), $R \geq 1$, there holds:*

$$\mathbb{E}\Big(f(\overline{w}_R) - f^*\Big) = \mathcal{O}\bigg(\frac{A_1}{\sqrt{RN}} + \frac{A_2}{R^{2/3}} + \frac{A_3}{R}\bigg),$$

where $\overline{w}_R = \frac{1}{R}\sum_{r=0}^{R-1} w_r$, with the problem related constants being

$$A_1 = \sqrt{\frac{\Delta_0}{N}\sum_{i=1}^{N}\left(\frac{\sigma_i^2}{q_iK_i} + \overline{q}_i\|\nabla f_i(w^*)\|^2\right)},$$

$$A_2 = \sqrt[3]{\frac{L\Delta_0^2}{N}\sum_{i=1}^{N}\frac{(K_i-1)}{q_i}\left(\sigma_i^2 + \|\nabla f_i(w^*)\|^2\right)},$$

$$A_3 = \Delta_0 \max\left\{\frac{64L}{q_{\min}}, 10\overline{K}L\right\}.$$

2) *For f μ-strongly convex, $R \geq \max\left\{\frac{64L}{q_{\min}\mu}, \frac{10L\overline{K}}{\mu}\right\}$, there exists a sequence of weights $\{v_r\}_{r\geq 0}$, such that there holds:*

$$\mathbb{E}\Big(f(\overline{w}_R) - f^*\Big) = \widetilde{\mathcal{O}}\left(\mu\Delta_0\exp\left(-\beta R\right) + \frac{B_1}{NR} + \frac{B_2}{R^2}\right),$$

where $\overline{w}_R = \frac{1}{V_R}\sum_{r=0}^{R-1} v_r w_r$, $V_R = \sum_{r=0}^{R-1} v_r$, with the problem related constants being

$$\beta = \min\left\{\frac{q_{\min}\mu}{128L}, \frac{\mu}{20L\overline{K}}\right\},$$

$$B_1 = \frac{1}{\mu N}\sum_{i=1}^{N}\left(\frac{\sigma_i^2}{q_iK_i} + \overline{q}_i\|\nabla f_i(w^*)\|^2\right),$$

$$B_2 = \frac{L}{\mu^2 N}\sum_{i=1}^{N}\frac{(K_i-1)}{q_i}\left(\sigma_i^2 + \|\nabla f_i(w^*)\|^2\right).$$

Here, $\Delta_0 = \big\|w_0 - w^*\big\|^2$ is the distance of the initial model from the true minimizer.

Several comments on Theorem 1 are now in order. First, the Theorem provides convergence rate guarantees for FedAvg for a more general heterogeneous setting than provided in existing studies such as [5], [7], [15] that assume equal number of local updates K_i's and uniform sampling probabilities q_i's. Next, the rates achieved for both smooth convex, and smooth strongly convex cases are order-optimal, i.e., when considering the slowest-decaying terms with respect to the number of epochs R, they scale order-optimally, as $1/\sqrt{R}$ and $1/R$, respectively. Furthermore, the established rates exhibit a linear speedup arising from FL in the number of users N. In more detail, with smooth convex functions, the leading error term scales as $1/\sqrt{NR}$. This rate is N-times faster than a central processing SGD method that would process at each iteration one of the functions f_i's sequentially. A similar conclusion also holds for the strongly convex case. Hence, we generalize the linear speedup effect established in [7] to heterogeneous settings. In addition, through constants $A_1 - A_3$ and B_1, B_2, we explicitly quantify the heterogeneity effects on performance, in terms of the σ_i's, K_i's, q_i's and $\|\nabla f_i(w^\star)\|$.

B. Optimal offline sampling probablities q_i

Note that the convergence rate established in Theorem 1 depends on the parameters q_i. In particular, the dominant terms associated with q_i are of the form $\sum_{i=1}^{N}\frac{c_i}{q_i}$,[2] where $c_i = \big\|\nabla f_i(w^*)\big\|^2 + \frac{\sigma_i^2}{K_i}$. Therefore, we would like to minimize

[2] Note that the communication probability associated with $\|\nabla f_i(w^*)\|^2$ is $\overline{q}_i = \frac{1-q_i}{q_i} = \frac{1}{q_i} - 1$. In the context of optimizing the communication probabilities, only the value $\frac{1}{q_i}$ is relevant, hence the expression $\sum_{i=1}^{N}\frac{c_i}{q_i}$ is correct.

quantity $\sum_{i=1}^{N} \frac{c_i}{q_i}$ with respect to probabilities q_i's. As it is expected, without constraints on the q_i's, the latter is minimized for $q_i = 1$, $i = 1, ..., N$. However, we impose two types of constraints on the q_i's. First, we have that $q_i \leq k_i$, where $k_i \in (0, 1]$ characterizes the capacity (quality) of the channel from user i to the server. Second, we let $\sum_{i=1}^{N} q_i = S$, for some $S < N$. This means that, on average, the server receives model increments from S users. This constraint mimics the uniform sampling scenario where a limited, small number of exactly S users are sampled at each epoch, in order to avoid the communication bottleneck at the server. This leads to the following optimization problem formulation:

$$\begin{aligned} \min_{q_1,\ldots,q_N} \quad & \sum_{i=1}^{N} \frac{c_i}{q_i} \\ \text{s.t.} \quad & 0 \leq q_i \leq k_i,\, i \in [N],\, \sum_{i=1}^{N} q_i \leq S \end{aligned}, \tag{4}$$

It can be shown (see [14]) that the solution is given by $q_i^* = k_i$, $i \in \mathcal{M}$, and $q_i^* = (S - k(\mathcal{M}))\, \frac{\sqrt{c_i}}{\sum_{j \notin \mathcal{M}} \sqrt{c_j}}$, $i \notin \mathcal{M}$, where $\mathcal{M}$ is the set of $|\mathcal{M}| = m$ indices corresponding to the largest values of c_i, $k(\mathcal{M}) = \sum_{i \in \mathcal{M}} k_i$, and m is either the largest positive integer satisfying $S - k(\mathcal{M}) \geq \frac{\sum_{i \notin \mathcal{M}} \sqrt{c_i}}{\sqrt{\tilde{c}_m}}$, or $m = 0$. Here, $\tilde{c}_m$ represents the m-th largest value of c_i's. We now comment on the q_i^*'s. We can see that, the larger the c_i, the larger $q_i^\star$ should be. Moreover, $q_i^\star$ is either greedily equal to k_i, or it is proportional to the value of $\sqrt{c_i}$. Consider quantity $c_i = \left\|\nabla f_i(w^*)\right\|^2 + \frac{\sigma_i^2}{K_i}$. We can see that, therefore, the server should sample less frequently the users with large K_i's. This is intuitive, as the users that make more local work reduce the local variance more and hence can transmit to the server less frequently. On the other hand, a user with a larger local SGD variance should communicate more frequently. This may appear counter-intuitive, but it is explained as follows: users with larger variance need to be reflected at the server side with more updates in order to filter out the local noise they incur. Finally, users with larger $\|\nabla f_i(w^\star)\|$ should communicate more frequently with the server. Intuitively, a user with a very small $\|\nabla f_i(w^\star)\|$ may safely skip its transmission when w_t is close to $w^\star$, because its contribution would not significantly change w_t.

C. Online adaptive communication protocol

Note that quantities c_i's depend on parameters that may be difficult to evaluate, such as $\|\nabla f_i(w^\star)\|$ and σ_i^2. These quantities can be estimated beforehand, so that a sub-optimal offline sampling probability scheme is devised by replacing the exact quantities with their estimates. For example, σ_i^2 relates to the local mini-batch size used by user i, while $\|\nabla f_i(w^\star)\|$ may be replaced with $\|\nabla f_i(w')\|$, where w' is a solution estimate available at the server. In alternative, an online adaptive user selection policy may be utilized. Reference [9] devises an online user sampling strategy assuming reliable communication links. When adapted to our framework and notation, this strategy works as follows. At epoch t, user i is selected with probability q_i^t, where q_i^t is the solution of (4) with quantity c_i replaced with $\|g_t^i\|^2$, and k_i set to one for all i. (We recall that $g_t^i = \frac{1}{K_i} \sum_{j=0}^{K_i-1} g_{t,j}^i$.) We next give a rationale that relates the optimal offline sampling derived here with the online strategy in [9]. When w_t is close to $w^\star$, we can approximate g_t^i as follows: $g_i^t \approx \frac{1}{K_i} \sum_{j=0}^{K_i-1} \left(\nabla f_i(w^\star) + n_{i,j}^t\right) = \nabla f_i(w^\star)$ $+\frac{1}{K_i} \sum_{j=0}^{K_i-1} n_{i,j}^t$, where $n_{i,i}^t$ is the user i SGD noise at the relevant inner and outer iteration. Taking the squared norm and expectation, while using independence of SGD noises across inner iterations, we can see that quantity $\mathbb{E}[\|g_i^t\|^2]$ can be approximated with c_i. In other words, when close to the solution, our offline sampling strategy is approximately the same as the adaptive strategy in [9]. The rest of the paper is devoted to a real-world implementation of the adaptive strategy in [9] on VCC tasks.

IV. Evaluation results

We now present implementation and evaluation studies of heterogeneous user sampling on real-world deployments for VCC. The main purpose of the section is two fold. First, we show that employing FL in VCC can improve model performance (accuracy) over the VCC models working in isolation. Second, we show that non-uniform user sampling can significantly improve FL communication efficiency.

We consider three training schemes; 1) each user (pilot site) trains the VCC model in isolation on its local data (this scheme is termed *local training*); 2) FL algorithm (2)–(3) with full participation, i.e., all users transmit to the server at all epochs (this scheme is termed here *FedAvg*); and 3) FL algorithm (2)–(3) with the optimal adaptive sampling strategy [9] (this scheme is termed here *NUS – Non-Uniform Sampling*). We implement each of the three training schemes using the *Flower* Federated Learning framework[3] for the Python programming language. With all schemes, all users receive the same initial model from the server which has its weights initialized to random values.

Each of the three pilot sites, namely, Valetta, Trento and Novi Sad, posses a local VCC input-output training dataset. For MT 170 videos and for GRN 660 videos were collected using static cameras. The corresponding data frames have been anonymized and subsequently annotated using the CVAT [4] tool. For the Novi Sad site, a dataset was collected within the staged recording that was carried out at the Petrovaradin fortress. No anonymisation was necessary as all human participants signed consent for the recording; we also note that in this type of experimental setup, identification of individuals is generally difficult due to the higher and overhead camera position. In total, around 800 annotated frames are provided for training. The model and loss functions adopted are as described in Section II-B. For the NUS policy, we set $S = 1$ (expected number of user transmissions per epoch). For both FedAvg and NUS, the total number of epochs is set to 100.

[3] https://flower.dev/

[4] https://cvat.org/

Table I shows the mean absolute error (MAE) at the three pilot sites at the end of training, for each of the three schemes. MAE was chosen for the optimization objective for this task (loss function). We can see that NUS improves performance over both local training and FedAvg for Novi Sad and Trento. On the other hand, the Valetta site incurs performance degradation. While monitoring the user selection process, we noticed that NUS rarely selects the Valetta user. Also, the distribution of the data samples in Trento and Novi Sad are quite similar, while for Valetta the corresponding distribution is different. For this reason, the Valetta site becomes "starved" of FL updates. In practice, the degradation due to FL can be easily fixed, by comparing the validation accuracy for both the joint FL model and the locally trained model and selecting the better performing model.

Figure 2 compares communication costs of FedAvg and NUS. We see the NUS cost reduction in model parameter transfers of around 17 gigabytes (17183MB). Therefore, NUS significantly reduces communication cost while at the same time improving MAE for the majority of sites. See also Figure 1 for evaluation of various losses versus execution time across the three sites for FedAvg and NUS.

TABLE I
FINAL VALIDATION MAE FOR LOCAL TRAINING, FEDAVG, AND NUS.

	NUS	**FedAvg**	**Local training**
Novi Sad	2.0890	3.1086	2.3622
Trento	27.6216	30.0928	40.3407
Valetta	9.1261	0.6900	0.6269

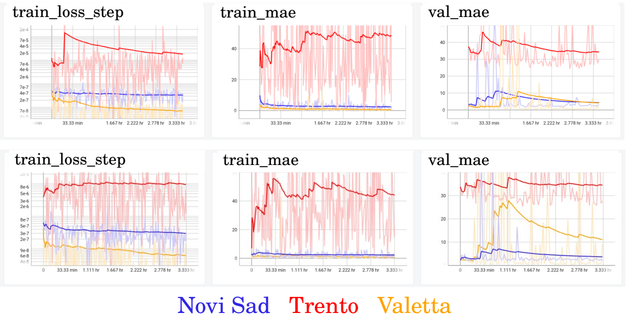

Fig. 1. Loss versus execution time. Top row: FedAvg; bottom: NUS. We monitor train_loss_step (training loss), train_mae (Training set MAE value) and val_mae (Validation set MAE value)

V. CONCLUSION

We considered a general framework for FL with various degrees of users heterogeneity, namely in terms of local gradient variances, activation probabilities, local number of updates, and (unreliable) link qualities. For this setting, we derived convergence rates for the FedAvg method, as well as optimal user activation probabilities that are pre-computed offline. We then connected these probabilties with previously derived optimal probabilities for an online setting [9]. A real-world data FL system for visual crowd counting across three sites (Valetta, Trento, Novi Sad) was then deployed and evaluated, featuring optimized user activation probabilities. The evaluation shows significant accuracy benefits of federation at poorly performing local sites, and significant communication savings due to the optimized, non-uniform sampling employed.

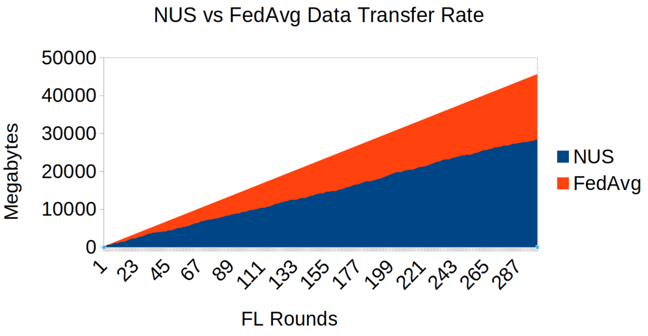

Fig. 2. Data transfers cost for FedAvg and NUS.

REFERENCES

[1] P. K. et al., "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, 2021.

[2] A. Armacki, D. Bajvovic, D. Jakovetic, and S. Kar, "Personalized federated learning via convex clustering," in *2022 IEEE International Smart Cities Conference (ISC2)*, Pafos, Cyprus, 2022.

[3] X. Cao, T. Başar, S. Diggavi, Y. C. Eldar, K. B. Letaief, H. V. Poor, and J. Zhang, "Communication-efficient distributed learning: An overview," *IEEE Journal on Selected Areas in Communications*, 2023.

[4] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *ICLR*, 2020.

[5] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[6] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," *ICLR*, 2021.

[7] Z. Qu, K. Lin, Z. Li, and J. Zhou, "Federated learning's blessing: Fedavg has linear speedup," in *ICLR 2021- DPML Workshop*, 2021.

[8] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "A novel framework for the analysis and design of heterogeneous federated learning," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5234–5249, 2021.

[9] W. Chen, S. Horváth, and P. Richtárik, "Optimal client sampling for federated learning," *Transactions on Machine Learning Research*, 2022.

[10] Y. Fraboni, R. Vidal, L. Kameni, and M. Lorenzi, "Clustered sampling: Low-variance and improved representativity for clients selection in federated learning," in *38th Internat. Conf. on Machine Learning*, 2021.

[11] Y. Jee Cho, J. Wang, and G. Joshi, "Towards understanding biased client selection in federated learning," in *25th International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.

[12] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Transactions on Signal Processing*, 2009.

[13] Q. Song, C. Wang, Y. Wang, Y. Tai, C. Wang, J. Li, J. Wu, and J. Ma, "To choose or to fuse? scale selection for crowd counting," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[14] A. Armacki, N. Milosevic, D. Bajovic, S. Kar, D. Jakovetic, A. Bakhtiarnia, L. Esterle, A. Muscat, and T. Festi, "Communication efficient model-aware federated learning for visual crowd counting and density estimation in smart cities," *preprint*, 2023. [Online]. Available: https://github.com/dusanjakovetic/FLVCC/blob/main/FLforVCCv2.pdf

[15] X. Gu, K. Huang, J. Zhang, and L. Huang, "Fast federated learning in the presence of arbitrary device unavailability," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021.